Spatio-temporal forecasting of network data

James Haworth

Department of Civil, Environmental and Geomatic Engineering,

University College London

Supervisors:

Prof. Tao Cheng
Prof. John Shawe-Taylor

A thesis sumbitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Geographic Information Science

October 2013

Statement of Originality

I, James Haworth confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed:

Date:

Abstract

In the digital age, data are collected in unprecedented volumes on a plethora of networks. These data provide opportunities to develop our understanding of network processes by allowing data to drive method, revealing new and often unexpected insights. To date, there has been extensive research into the structure and function of complex networks, but there is scope for improvement in modelling the spatio-temporal evolution of network processes in order to forecast future conditions.

This thesis focusses on forecasting using data collected on road networks. Road traffic congestion is a serious and persistent problem in most major cities around the world, and it is the task of researchers and traffic engineers to make use of voluminous traffic data to help alleviate congestion. Recently, spatio-temporal models have been applied to traffic data, showing improvements over time series methods. Although progress has been made, challenges remain. Firstly, most existing methods perform well under typical conditions, but less well under atypical conditions. Secondly, existing spatio-temporal models have been applied to traffic data with high spatial resolution, and there has been little research into how to incorporate spatial information on spatially sparse sensor networks, where the dependency relationships between locations are uncertain. Thirdly, traffic data is characterised by high missing rates, and existing methods are generally poorly equipped to deal with this in a real time setting.

In this thesis, a local online kernel ridge regression model is developed that addresses these three issues, with application to forecasting of travel times collected by automatic number plate recognition on London's road network. The model parameters can vary spatially and temporally, allowing it to better model the time varying characteristics of traffic data, and to deal with abnormal traffic situations. Methods are defined for linking the spatially sparse sensor network to the physical road network, providing an improved representation of the spatial relationship between sensor locations. The incorporation of the spatio-temporal neighbourhood enables the model to forecast effectively under missing data. The proposed model outperforms a range of benchmark models at forecasting under normal conditions, and under various missing data scenarios.

Acknowledgements

This thesis would not have been possible without the fantastic support of the people around me. First and foremost, I thank my supervisor, Prof. Tao Cheng. Throughout the past 4 (and a bit) years she been unwavering in her support, and has given up a huge amount of her time helping me to make the best of my research. She has always believed in my ideas and provided me with the motivation to continue, helping me through the more difficult times and joining in the better times. I also thank my second supervisor, Prof. John Shawe-Taylor, for his valuable contribution to my work. Spending one hour talking with him is worth a month or more buried in books.

I also owe a debt of gratitude to the members of the STANDARD project; Andy Emmonds, Jonathan Turner and Alex Santacreu at Transport for London; and Prof. Benjamin Heydecker and Dr. Andy Chow at UCL. Your experience and knowledge cannot be matched, and has been invaluable to me.

I would also be nowhere without the support, advice and friendliness of my colleagues at UCL. Jiaqiu Wang helped me find my feet when I was new to the PhD life, and I have enjoyed collaborating with him over the past four years immensely. My friends at UCL, Ed Manley, Adel Bolbol, Garavig Tanaksaranond, Berk Anbaroglu, Ioannis Tsapakis, Jingxin Dong, Monsuru Adepeju, Leto Peel, Suzy Moat and Ryan Davenport have helped make these some of the most enjoyable years of my life, despite the challenges we have all faced together.

I also thank the Engineering and Physical Sciences Research Council (Grant EP/G023212/1) for their financial support, and Transport for London for providing the data used in this thesis.

Finally, I thank my family and friends outside of UCL, especially my parents, Christine and Malcolm, for putting a roof over my head for the first three years of my PhD (and many before), and Amy, for putting up with a PhD student for so long, which is no easy task. The fact they all remember who I am after the past year shows what caring people they all are.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Progress in network science

Networks are everywhere and human beings interact with them on a daily basis; commuting between home and work on transportation networks, accessing information on the World Wide Web over the Internet, communicating with others over mobile phone networks; all the while using the neural network in their brain to accomplish these tasks. Networks provide a rich metaphor for many real world processes, systems and structures, both physical and abstract.

Complex networks are those networks whose structure is irregular, complex and dynamically evolving in time (Boccaletti et al., 2006). Since the latter period of the 20th century, there has been an explosion of research into the structure and function of complex networks. This has evolved into the field of *network science*, which has produced significant scientific breakthroughs in terms of network structure, notably, many networks exhibit small world (Watts and Strogatz, 1998) and scale free (Barabási and Albert, 1999) properties. For example, in the study of road networks, a great deal of recent research has focused on topology, notably the influence configuration and connectivity hold on network usage and dynamics (Hillier and Iida, 2005; Castells, 2010; Pflieger and Rozenblat, 2010). Small world phenomena are also observed on transportation networks due to the interplay between network structure and the dynamics taking place on them (Xu and Sui, 2007). Jiang (2007) observes that the topologies of urban street networks based on street-street intersection exhibit scale free properties and follow a power law distribution, with 80% of streets having length or degree less than average and 20% having length or degree greater than average. This property has been shown to apply to traffic flow on street networks (Jiang, 2009).

Yet such research, although valuable in understanding the structure and properties of networks, fails to fully explain the nature of the spatio-temporal dynamics occurring on the network. Network processes are often characterised by spatio-temporal patterns of nonstationarity, nonlinearity and heterogeneity. The nonstationarity usually refers to the trends and seasonal/cyclic patterns, such as morning-peak and seasonal change of traffic flow on road network. The nonlinearity of spatio-temporal patterns could be in various forms, particularly heteroskedasticity, which is nonconstant variance in time and/or space (Stathopoulos and Karlaftis, 2001; Cheng et al., 2011; Cheng et al, 2013). Nonlinearity in

space is commonly referred to as heterogeneity, which is a term that encompasses the notion that the earth's surface displays incredible variety (De Smith et al., 2007).

All these issues make the forecasting of network performance extremely challenging. Recently, models have been developed, for example, to simulate the spread of, and immunization against disease (Pastor-Satorras and Vespignani, 2002; 2005; Cohen et al., 2003), in both human populations and technological networks. But more efforts are needed on this front for many real world networks such as transport, electrical grids (U.S. Dept. of Energy, 2009) and the Internet. More importantly, with real time data collected on many real world networks, it is the time to move from structural analysis and simulation to the modelling and forecasting of these data.

1.2 Forecasting of network data

Forecasting models of network data have traditionally focussed on time series approaches. In such approaches, the network structure is ignored and each of the time series at the nodes is forecast independently. Examples of such models exist in electrical load forecasting (Alfares and Nazeeruddin, 2002; Hahn et al., 2009), Internet traffic forecasting (Basu et al., 1996; Papagiannaki et al., 2003) and road traffic forecasting (Vlahogianni et al., 2004), amongst others. While time series approaches can achieve excellent results, they ignore the spatio-temporal evolution of the network process, and often perform poorly under abnormal conditions and when data are missing (Vlahogianni, 2007).

More recent approaches use network structures to construct spatio-temporal models that capture the spatio-temporal dependency between locations where network data are observed. The traditional way to incorporate spatial structure into a forecasting model is to assume that the spatio-temporal relationship can be described by a linear model with global set of parameters, which requires the space-time process to be weakly stationary (Cressie and Wikle, 2011). However, such models typically cannot deal with the nonstationary and nonlinear properties of network data, especially when network data are highly dynamic (Kamarianakis and Prastacos, 2005; Cheng et al. 2013).

In response to this, statistical modelling frameworks have been extended to model local and/or dynamic space-time structures recently (Min et al., 2009a; Ding et al., 2010; Min and Wynter, 2011; Kamarianakis et al., 2012; Cheng et al, 2013). The improvement that these models gain over global model specifications is striking, and clearly motivates the use of

local model structures (Cheng et al, 2013). However, although these approaches are well grounded in statistical theory and show promising performance, the configuration of the spatial and temporal structure for such models is non-trivial, which makes their computation costly, and limits their application to large networks.

Increasingly, researchers and practitioners are turning towards less conventional techniques, often with their roots in the machine learning (ML) and data mining communities, that are better equipped to deal with the heterogeneous, nonlinear and mutli-scale properties of large scale network datasets. For instance, many of the ML methods that have recently proliferated in the time series forecasting literature have shown strong performance (Ahmed et al., 2010), particularly in the context of longer range forecasts (Vlahogianni et al., 2004). It has been demonstrated, for example, that a globally trained spatio-temporal artificial neural network (ANN) model can outperform a linear state-space model with a local space-time structure (Vlahogianni et al., 2007). Of these methods, kernel methods are particularly attractive as they combine the advantages of globally optimal linear models with nonlinear capabilities. However, these approaches mainly use globally fixed structures. How to combine the local spatio-temporal structure in ML models in order to improve their forecasting accuracy and explanatory power is a challenge for space-time forecasting of network data.

1.3 Challenges in modelling urban road networks

This thesis focusses on a particular type of spatial network (Barthélemy, 2011), the urban road network. The smooth operation of urban road networks is fundamental to the health of cities worldwide. Poorly functioning road networks have severe economic, social and environmental costs related to traffic congestion, environmental pollution and public safety (Arnott and Small, 1994; Walters, 1961; Goodwin, 2004; Barth and Boriboonsomsin, 2008). Because of their importance, many urban road networks are equipped with sensor networks that observe the traffic stream for a variety of purposes such as enforcement, incident detection and traffic monitoring (Chong and Kumar, 2003). On a sensor network, a time series of observations of the network process is collected at each node of the network. These data are used in forecasting systems to aid urban traffic control systems (UTCS), advanced traffic management systems (ATMS) and advance traveller information systems (ATIS) (Vlahogianni et al., 2004).

To make best use of this data, it is necessary to take a network-based approach, which enables modelling of the spatio-temporal evolution of traffic conditions. If one can model the way in which the current process evolves from its past values, then one can model causation, which is the "holy grail of science" (Cressie and Wikle, 2011, p.297). In other words, one can model the spread of congestion. This is particularly important in the context of non-recurrent events, whose spatio-temporal effect can be large. Despite this, the uptake of spatio-temporal models in the traffic forecasting literature has been slow, and there are very few models that have been developed for large scale forecasting of urban road networks (Yue and Yeh, 2008). Historically, the main reason for the slow uptake and development of spatio-temporal models has been the lack of available data and computational power (Griffith, 2010). This situation has now been reversed, and replaced with the problems of modelling so called *big data* (Lynch, 2008; Boyd and Crawford, 2011; Manyika et al., 2011). In the context of traffic data, these problems can be summarised around three points:

Firstly, large scale network data are often subject to high levels of noise and missing data that must be dealt with on the fly (Zhong et al., 2004a; Griffith, 2010; L. Li et al., 2013). Missing data periods are often lengthy and not randomly dispersed in the data (Qu et al., 2009). Missing data are usually ignored in the context of forecasting, and dealt with offline using imputation (Zhong et al., 2004a). This is insufficient to maintain forecasting performance in real time. A small number of recent studies have attempted to address this by accounting for missing data in the forecasting algorithm (Whitlock and Queen, 2000; van Lint et al., 2005; Chan et al., 2013b), but they are applied to highway data where the spatio-temporal dependency relationships are more certain. Application to urban networks has not been considered, despite being a serious problem.

Secondly, due to the prohibitive cost of installing and maintaining sensor networks, as well as processing and storing data collected using them, sensors are often sparsely located in space (Liu, 2008), and do not have sufficient spatial coverage to model the physical relationship between locations in an unambiguous manner. The idealised representation of the network structure that one would expect is often not mimicked in the sensor network structure, and must be elicited (Haworth et al., 2013). Furthermore, one typically has to develop models from sensor networks that were not constructed with forecasting in mind. Often the spatio-temporal relationships between locations are nonlinear, heterogeneous and weak, and difficult to model using conventional methods (Cheng et al., 2011).

Finally, the sheer amount of data being collected means that network models must be capable of processing incoming data and producing forecasts in a timely fashion. This leads to a trade-off between the complexity of the process that can be modelled and the computational efficiency of the algorithm. This is a difficult task, and consequently the spatio-temporal models of urban networks described in the literature are typically applied to small networks, which are not indicative of the networks of entire cities. Linked to this problem is the issue that network processes are dynamic, and models must be able to adapt to changes in the data distribution over time in order for performance to be maintained. Such models are known as online, and have only been developed in a small number of studies to date in the context of freeway data (Castro-Neto et al., 2009; van Lint, 2008). Application of online algorithms to urban road networks has not been attempted.

1.4 Research objectives

In light of the challenges described in the preceding sections, the aim of this thesis is to develop a forecasting model of network data that fulfils the following research objectives:

Objective 1 – To develop a framework for modelling the spatio-temporal structure of real world network data.

Objective 2 – To integrate the spatio-temporal structure of networks into a forecasting model that is capable of operating in real time, and is robust to the presence of missing data.

Objective 3 – To apply the model to the case of forecasting travel times on an urban road network, where the sensor network is sparsely deployed on a dense road network.

Objective 4 - To develop an evaluation framework for assessing the strengths and weaknesses of the proposed model.

1.5 Scope of the thesis

There are many potential sources of traffic data on London's road network. Many of the signalised intersections are controlled by Split Cycle Offset Optimisation Technique (SCOOT), and this system collects flow, delay, congestion, detector occupancy and detector flow (Imtech, 2013). Transport for London also maintains an incident database called the London Transport Information System (LTIS), which has details of planned and unplanned road works, incidents etc. Aside from this, floating vehicle data obtained from mobile phone and

GPS devices are becoming more widespread. In each of these data sources there are non-trivial issues with data processing and/or acquisition that complicate their use.

This thesis concentrates on the short term forecasting of travel time data collected using automatic number plate recognition (ANPR) data as part of Transport for London's (TfL) London Congestion Analysis Project (LCAP). In this context, short term forecasting is taken to mean forecasting of intra-day traffic. The application to long term forecasting of traffic variables for planning purposes is not considered, although it is an important problem in its own right.

In accordance with the scope, the literature review section of this thesis is focussed primarily on methods for forecasting road network data. However, the proposed model is not tailored to the specific problem of forecasting road network data. Extensions to the model framework are given in the concluding chapter that suggest ways of applying the model in a wider context.

1.6 Structure of the thesis

This thesis is organised into nine main chapters and two appendices that contain supplementary material and additional results that are not presented in the main body of the thesis. A roadmap of the thesis structure is shown in figure 1.1. The content of the chapters is as follows:

In Chapter 2, entitled *Networks*, the basic concepts and terminology of networks are introduced. Particular attention is paid to spatial networks in order to limit the scope of the review. Methods for forecasting network data are introduced.

Chapter 3, entitled *Spatio-temporal forecasting of road network traffic* reviews the progress to date on modelling and forecasting of traffic data collected on road networks. The chapter focusses on three facets of the models described in the literature: 1) the types of model that have been used, which are broadly separated into parametric and machine learning approaches; 2) the ways in which spatio-temporal information are incorporated into the models, with particular attention paid to those models that are applied to urban road networks, and; 3) the ways in which missing data are dealt with in forecasting models. The findings of this chapter motivate the direction of research in chapters 4, 5 and 6.

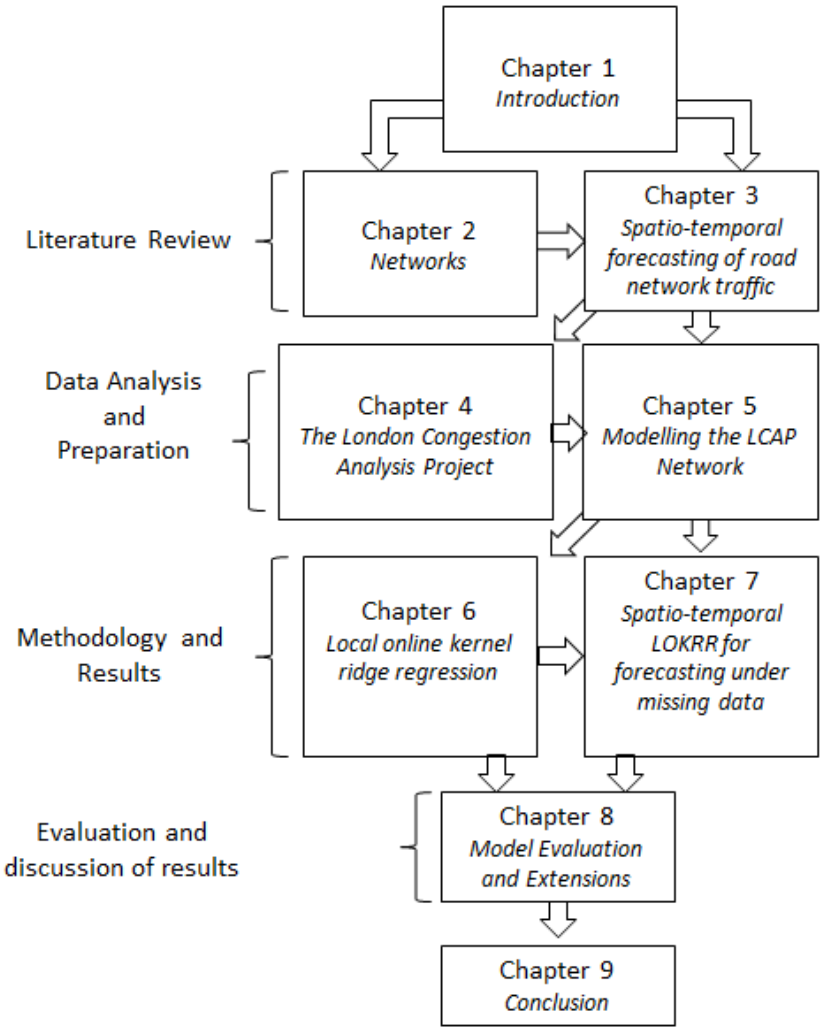


Figure 1.1 – Thesis roadmap

In Chapter 4, entitled *The London Congestion Analysis Project*, the case study data is introduced, which is the London Congestion Analysis Project (LCAP). The LCAP is a network of automatic number plate recognition (ANPR) cameras that collect travel time data on London's road network. A thorough missing data analysis is presented.

Chapter 5, entitled *Modelling the LCAP Network*, describes how the LCAP network can be modelled using a spatial weight matrix. The sensor network is fused with the physical road network in order to produce a better representation of the spatio-temporal relationship between sensor locations. A space-time autocorrelation analysis is carried out that reveals a pattern of strong temporal autocorrelation and weak spatio-temporal autocorrelation. Based on these findings, and the findings of Chapters 3 and 4, some requirements of a space-time model of network data are defined.

In Chapter 6, entitled *Local Online Kernel Ridge Regression,* a temporally local forecasting model is described for forecasting of seasonal time series data. The model is kernel based, in which small kernels are defined for each time of day, each with their own parameters. The model is online and is able to accommodate new data as they arrive. Although applied to traffic data, the model is applicable to any time series that displays cyclic variation, and can be extended to other types of time series.

In Chapter 7, entitled *Spatio-temporal LOKRR for forecasting under missing data* the LOKRR model is extended to incorporate the spatial structure of networks. The spatio-temporal (ST)LOKRR model is applied to forecasting of travel times where data are assumed to be missing due to sensor failure.

In Chapter 8, entitled *Model Evaluation and Extensions*, the model evaluation framework is described, against which the performance of the LOKRR and STLOKRR models is evaluated. Based on the results of the evaluation, some model extensions are briefly described that will form part of the basis for future research.

Finally, in Chapter 9, entitled *Conclusion*, the thesis is concluded. The main outcomes of the research are stated, and the contributions to the scientific literature are elucidated. The thesis ends with a statement about the future directions for research in the subject area.

Chapter 2 Networks

2.1 Chapter overview

This chapter charts the progress to date in modelling networks, with particular attention paid to spatial networks. In section 2.2, networks are defined, and a brief introduction to complex networks and the field of network science is given. The aim of this section is not to provide an exhaustive survey of the network science literature, but to alert the reader to some of the major achievements of network science. Section 2.3 contains a discussion of network processes, and how they can be incorporated into mathematical models. Section 2.4 introduces spatial network structures, and how they differ from other networks, before the properties of spatio-temporal processes are introduced in section 2.5. In section 2.6, parametric and machine learning (ML) approaches to modelling spatio-temporal network data are described, and the main approaches are reviewed. Finally, the Chapter is summarised in section 2.7.

2.2 Background: Networks, complex networks and network science

A network is a set of discrete items, called ***nodes***, with connections between them, called ***edges***. A network is typically represented as a graph $G = (N, E)$, with a set of N nodes and E edges. Networks can be broadly separated into four categories, which are social networks, information networks, biological networks and technological networks (Newman, 2003). Table 2.1 summarises these types of network and some basic terminology of graphs is given in table 2.2. These terms are used throughout this thesis.

Since the 1990s, much research on networks has focused on the theory of ***complex networks***, which are those networks whose properties are irregular, complex and dynamically evolving in time (Boccaletti et al., 2006). The study of such networks has been facilitated by increases in computing power and the availability of massive datasets pertaining to real life networks, as well as the emergence and growth of new types of networks in the digital world such as the Internet and online social networks.

Ground-breaking research has revealed that certain structural properties pervade many types of network, notably the small world property (Watts and Strogatz, 1998) and the scale-free property (Barabási and Albert, 1999). A scale free network is one whose degree

distribution follows a power law. Many real world networks have been shown to be scale free, including the World Wide Web and social networks.

Table 2.1 – Types of Network (Summarised from Newman, 2003)

Type of network	Description	Examples
Social Network	A set of people (or animals) or groups of people (nodes) with associations between them (edges).	Friendship networks Academic collaborations Business relationships Animal behaviour
Information Network	Networks of information flow. Pieces of information are stored at the nodes. The edges between nodes signify connections between information sources.	Academic citation networks. World Wide Web
Biological Network	Networks of biological systems, where the nodes are biological entities, and the edges are connections between them.	Metabolic Networks Neural Networks Food Webs
Technological Network	Man-made, physical networks, designed to transport people, resources or commodities.	Transportation networks (Roads, rail, airlines) The Internet Electricity Grid

In a scale free network, a small number of nodes are highly connected and have huge importance within the network. In a small world network, the typical distance between two nodes grows proportionally with the logarithm of the number of nodes in the network, which essentially means that a path between any two nodes in the network will typically consist of a small number of steps. Small world networks are themselves scale free. The discovery of these properties has been tremendously important in shaping our understanding of networks, and has led to important research into the resilience of networks (Dobson et al., 2007), the growth of networks (Albert and Barabási, 2002) amongst other things.

The surge in research has led to various works being published that summarise the progress to date in the study of complex networks and attempt to formalise the concepts in terms of graph theory (see, for example, Albert and Barabási, 2002; Newman, 2003; Boccaletti et al., 2006). The early work formed the building blocks of the field of network science, which has now become a mature subject in its own right, exemplified by the creation in 2013 of the *Network Science* journal by Cambridge University Press (Brandes et al., 2013). The network science literature provides a diverse range of tools and techniques for analysing and enhancing our understanding of the structure and large scale statistical properties of networks.

Table 2.2 – Basic terminology of graphs

Term	Definition
Directed Edge	An edge with an associated direction. It is directed from its *start node* to its *end node*
Directed graph (digraph)	A graph in which all the edges are directed
Undirected graph	A graph that contains no directed edges
Adjacency	Two nodes in a graph are considered to be adjacent if they are connected by an edge. Likewise, two edges in a graph are considered to be adjacent if they share a common node.
Incidence	If node *n* is an node of an edge *e*, then *n* is incident on *e*, and vice versa.
Node degree	The degree of a node *n* in graph *G*, denoted deg(*n*), is the number of edges incident on *n* plus twice the number of self-loops. Self-loops are not considered in this thesis.
Weighted graph	A graph in which each edge is assigned a number, called an edge weight.
Path	A sequence of edges that connects a sequence of nodes in a graph.

2.3 Modelling Network Processes

The processes that take place on networks can be of equal interest to the structure of networks themselves. For example, in social networks, the nodes and edges themselves are of great interest because they reveal the structural properties of social relationships.

However, it is difficult to understand *why* such properties exist without examining the *attributes* of the nodes. Looking at the attributes allows one to answer questions such as: *what characteristics make a certain node more connected than another node?* And; *what characteristics do connected nodes share*? In complex networks, it is a reasonable assumption that nodes that are connected by an edge will be similar in some way. More specifically, one or more of their attributes will be correlated.

It is the interplay between the processes occurring on networks and the complexity and dynamics of the network structure where there is great potential for research. The ultimate goal of the study of the structure of complex networks is to gain an understanding of the systems built upon them (Newman, 2003). For example, one may wish to explore how the structure of a social network affects the spread of information in times of crisis or political unrest such as the Arab Spring that began in 2010 (Lotan et al., 2011; Khondker, 2011) or the London riots of 2011 (Tonkin et al., 2012); or one may wish to know how network structure causes congestion and cascading failures in flow networks (Albert et al., 2000; De Martino et al., 2009; Holme, 2003; Rosas-Casals et al., 2007; Zhang and Yang, 2008).

2.3.1 Adjacency matrices

To analyse and model a network process, a network can be represented mathematically using an adjacency matrix. Given an undirected graph G with N nodes and E edges, an adjacency matrix A is an $N * N$ square matrix. Its i, j element is 1 if nodes i and j are adjacent. That is, $a_{i,j} = 1 \; iff \; (n_i, n_j) \in E$. For an undirected network, the adjacency matrix is symmetric, and its leading diagonal is zero provided it does not contain any loops (edges with the same start and end node)[1]. Adjacency matrices are convenient as they can be used directly in correlation measures and regression models. For instance, a network autoregressive model can be written as:

$$\mathbf{y} = \rho \mathbf{A}\mathbf{y} + \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim N(0, \sigma^2) \tag{2.1}$$

[1] Alternatively, one may define an incidence matrix, which is an $N * E$ matrix. In its simplest form, it is a binary matrix where the i, j element is 1 if node i is incident on edge j, 0 otherwise.

Where ρ is the network lag coefficient, $\mathbf{y}$ is the vector of dependent variable values and $\boldsymbol{\varepsilon}$ is the vector of error terms (Peeters and Thomas, 2009). Here, the adjacency matrix is usually row standardized so that all rows sum to one, which has the effect of including a weighted sum of the values at adjacent nodes in the regression equation (De Smith et al., 2007). In a network autoregressive model, the dependent variable appears on both sides of the equation because $\mathbf{y}$ is modelled as a linear combination of lagged values of itself, recorded at adjacent nodes in the network.

This type of model is common in the field of network autocorrelation analysis, which emerged independently of the complex networks literature in the late 1970s. Network autocorrelation borrows techniques from spatial analysis and geography as tools for analysing data collected on networks. Network autocorrelation models aim to model the relationship between variables measured at the nodes of a network. They were first applied to social and cultural data. Dow et al. (1982) extend the spatial autocorrelated disturbances (errors) model to the network case by changing the structure of the spatial weight matrix to reflect network linkages. In this type of model, the errors are not independent and are considered as a weighted average of the errors at related units, determined by the weight matrix (Dow et al., 1984).

Various extensions to this method have been devised and the effect of different network structures on spatial and network autoregressive models is examined in Farber et al. (2009). A diverse range of tangible and intangible networks have been modelled including social networks (Doreian et al., 1984; Dow et al., 1984; Dow, 2007; Dow and Eff, 2008; Leenders, 2002), migration networks (Black, 1992; Chun, 2008) and transportation networks (Black and Thomas, 1998; Flahaut et al., 2003).

2.4 Spatial Network Structures

A spatial network is a particular type of network whose nodes are embedded in Euclidean space, and whose edges are real physical connections (Boccaletti et al., 2006). Spatial networks have unique characteristics that require different treatment to aspatial networks. The principal influence on the structure and function of spatial networks is the physical distance between nodes. On spatial networks, the presence of an edge between two nodes is limited by their physical separation distance. For example, in the network of airports, a route (edge) can only exist between two airports (nodes) if their separation distance does not exceed the maximum range of aircraft.

Spatial networks can be planar. A planar network is one whose edges do not intersect when drawn in the plane (Barthélemy, 2011). In planar spatial networks, such as road networks and river systems, the degree of a node is constrained by the physical space available for edges (Boccaletti et al., 2006). For instance, in the planar representation of a road network, the number of road segments that join at an intersection is constrained by the space available for roads. It is uncommon to encounter intersections that connect more than 5 or 6 road sections. Hence, planar networks necessarily cannot be scale free (Cardillo et al., 2006)[2]. The airport network is a spatial network that is not planar as routes may cross one another.

2.4.1 Spatial weight matrices

The network metaphor has been used to represent geographical space in quantitative geography for over 40 years (Barthélemy, 2011). The quantitative modelling of spatial processes using a network representation is achieved using spatial weight matrices. Given a network of spatial units $G(S, E)$, with S spatial units and E edges connecting them, a spatial weight matrix **W** is an $S * S$ square matrix in which the element $w_{i,j} > 0 \; iff \; (n_i, n_j) \in E$. The difference between **W** and an adjacency matrix is that its elements need not be binary, meaning that it can contain a much richer representation of the geographic space. Cliff and Ord's (1969) seminal work demonstrates that a weight matrix representing various types of spatial connections can better account for the nuances of real world spatial dependencies than a binary matrix (Getis, 2009).

A non-binary weighting scheme in **W** equates to representing the spatial units as a weighted graph. The choice of the weights is application specific and allows one to incorporate prior knowledge of the spatial process into a model. For example, in the case of areal data the weights may be specified as the length of the shared border, the proportion of shared perimeter or the number of neighbours (Getis and Aldstadt, 2004). If **W** is symmetric, the relationship between locations is assumed to be identical in both directions, meaning that the perceived influence of unit i on unit j is equal to the perceived influence of unit j on unit i. If **W** is asymmetric then direction is introduced to the system, which equates to $G(S, E)$ being a directed graph. Directed graphs are used in systems of flows, such as transportation networks.

[2] Road networks need not be represented as planar and can be based, for example, on street-street intersection, in which case they can be scale-free (Jiang, 2007).

Determining the weighting structure and the value of the weights in **W** is a non-trivial issue. Different weight matrices can lead to different inferences being drawn and can induce bias in models. The effect of this has been explored in the spatial literature (see Stetzer, 1982; Florax and Rey, 1995; Griffith and Lagona, 1998) and the networks literature (Mizruchi and Neuman, 2008; Páez et al., 2008; Farber et al., 2009; Neuman and Mizruchi, 2010). On spatial networks, network distance between neighbouring units can be used as the weighting scheme, although this is not always straightforward. Direction of dependence is often important, and may change with time, and the dependence may not decay linearly with spatial distance.

2.5 Spatial Network Processes

There are many different types of process that operate on spatial networks. This thesis is concerned primarily with flow networks, in which some quantity flows between the nodes of the network along the edges. Examples of flow networks are transportation systems, the Internet and electrical grids. On flow networks, observations of the network process are (usually) made at regular intervals using sensors, which form time series. These sensors are distributed across the network, and when considered together form a space time series that describes the network process. The processes that are observed on spatial networks typically exhibit characteristics that require special treatment in space-time modelling frameworks. In the following sections, the temporal and spatial characteristics of spatial networks are described in turn.

2.5.1 Time series

Observations of a space-time process are made at the nodes of a spatial network, which are sampling locations such as weather stations, traffic flow detectors or census output areas. At each node an individual time series is collected that describes the temporal evolution of the space-time process at a single point in space. A time series $z = \{z(t)|z \in T\}$ in temporal domain T is a sequence of observations of a process, usually taken at equally spaced time intervals; $z_t, z_{t-1}, z_{t-2}, \ldots z_{t-(n-1)}$, where z_t denotes the value of some quantity z such as traffic flow volume at time t and n is the length of the series. $z_{t-(n-1)}$ indicates the time at which the first available observation was made.

2.5.1.1 Stationary process

A strictly stationary process is one whose joint probability distribution does not change with time, which entails that the joint probability distribution of a series with n observations $z_t, z_{t-1}, z_{t-2}, \dots z_{t-n}$ made at any set of times $t, t-1, t-2, \dots, t-(n-1)$ is the same as that associated with n observations recorded at another set of times $t+k, (t-1)+k, (t-2)+k, \dots, (t-(n-1))+k$, for any choice of k, which is known as the temporal lag (Box et al., 1994, p.24). A strictly stationary process has constant mean and variance. The joint probability distribution $p(z_t, z_{t+k})$ is also the same for all values of k, meaning that the autocovariance between z_t and z_{t+k} is equal for all t.

2.5.1.2 Temporal dependence and autocorrelation

An observation from nature is that near things tend to be more similar than distant things in time. For instance, the weather tomorrow is more likely to be similar to today's weather than the weather a week ago, or a month ago and so on. This phenomenon is known as temporal dependence. The presence of temporal dependence violates the stationarity property described in section 2.5.1.1 that is a requirement of classic statistical models such as ordinary least squares (OLS). Temporal dependence can be measured using the temporal (or serial) autocorrelation function[3] (ACF, see, for example, Box et al., 1994). Autocorrelation is the cross-correlation of a series with itself. It is calculated at discrete temporal lags and, for interpretability, is usually displayed as an ACF plot with confidence intervals shown (see figure 2.1).

If a time series exhibits autocorrelation that does not decay exponentially toward, and remain at, zero then it is nonstationary, meaning it does not have constant mean and variance. If the autocorrelation decays toward zero initially but returns to significant positive values at regularly spaced intervals, then the series contains a seasonal (cyclical) pattern. This type of pattern is common in many series, including traffic data where it is caused by rush hour traffic patterns. Many linear time series models, such as the autoregressive integrated moving average (ARIMA) model, require removal of autocorrelation from the series prior to model building in order to be effective. This is usually accomplished through seasonal or nonseasonal differencing (Kendall and Ord, 1990, p.37-39).

[3] The autocorrelation function is described in Chapter 5.

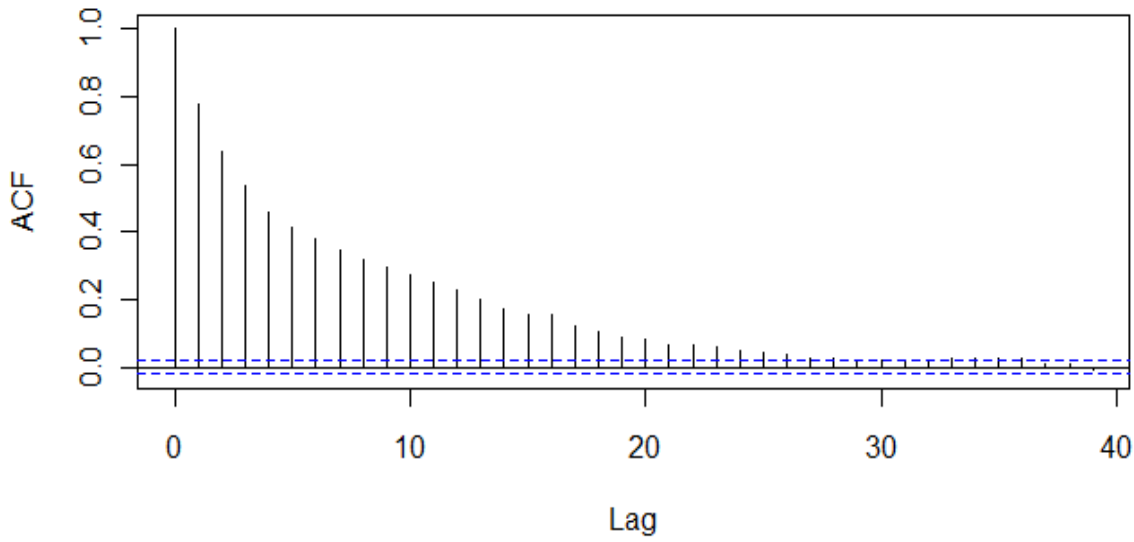


Figure 2.1 – A typical ACF plot showing strong, significant autocorrelation, decaying exponentially toward zero. Approximate 95% confidence intervals are shown as blue, horizontal lines. Note that the autocorrelation at lag zero is always 1.

2.5.1.3 Nonlinearity

Some time series exhibit nonlinear characteristics. The term nonlinear is somewhat ambiguous; for instance, if a series can be explained by a simple analytic function such as a power, sinusoid or exponential then, although nonlinear, it can often be linearized and modelled with linear techniques (Cook, 1987). Hence, series of this type are usually not considered to be nonlinear in the time series literature. Here, nonlinear is taken to mean those time series that cannot be modelled using linear techniques. An example of a nonlinear series is a cyclic time series that increases at a faster rate than it decreases, or vice versa (Chatfield, 2004). In such series, future values of the series do not depend linearly on previous values. Nonlinear series of this type cannot be explained by a linear model and may benefit from a nonlinear approach. There are various methods for testing for nonlinearity in time series which are reviewed in Tsay (1986). Lee et al. (1993) compare a range of the available methods. In the presence of nonlinearity, a range of nonlinear time series techniques have been developed including bilinear models (Granger, 1978) and machine learning techniques such as Artificial Neural Networks (ANNs) ANNs and Support Vector Regression (SVR).

2.5.1.4 Heteroskedasticity

One very common form of nonlinearity that is common in real time series is heteroskedasticity. Heteroskedasticity is present in a time series when its variance is not constant over time. This has been recognised as a particular problem in financial series where volatility clustering is common (Mandelbrot, 1963). It is also present in flow networks, where variance is higher in peak periods. Like temporal dependence, this property violates the assumption of stationarity; and leads to consistent but inefficient parameter estimates and inconsistent covariance matrix estimates (White, 1980). The autoregressive conditional heteroskedasticity (ARCH) model of (Engle, 1982) was developed specifically to deal with this phenomenon. As with an ARIMA model, it is applied to uncorrelated, zero mean time series but the conditional variance is modelled as nonconstant. Engle won the Nobel Prize in 2003 for his contribution, and the basic method has since been widely applied, modified and extended and Engle (2001) provides a summary. Traffic series are heteroskedastic as the variance is higher during congested periods (Fosgerau and Fukuda, 2012).

2.5.2 Spatial characteristics

When a variable z is observed at two or more locations it can be considered a spatial series $z = \{z(s)|s \in S\}$ in spatial domain S. At each time t, the series of observations $z_1, z_2, \dots, z_s$ collected at the nodes of a spatial network forms a spatial series. Spatial series share many of the characteristics of time series, but they are fundamentally different because space is two or three dimensional whereas time is one dimensional and unidirectional.

2.5.3 Spatial stationarity

A stationary spatial process is one whose mean and variance are constant in space. The covariance between observations of the processes depends only on their separation distance (Cressie, 1988). Firstly:

$$E[z(s)] = \mu \quad \forall\, s \in S \tag{2.2}$$

Secondly:

$$cov[z(s_1), z(s_2)] = C(s_1 - s_2) \qquad \forall\, s_1, s_2 \in S \tag{2.3}$$

If $C(s_1 - s_2)$ is a function only of the distance between s_1 and s_2 then the process is also isotropic, which means that the variance is constant in all directions.

2.5.4 Spatial dependence and autocorrelation

Like temporal data, spatial data often violate the normality assumptions required by standard statistical models because they exhibit spatial dependence. This phenomenon is neatly summed up neatly by Tobler's famous first law of geography:

> *"Everything is related to everything else, but near things are more related than distant things"* (Tobler, 1970).

To give an example, if one were supplied with some precise GPS coordinates in middle of the Sahara desert, one may infer with some degree of certainty that the terrain type would be desert. To make this inference, one would only require knowledge of the Sahara and nowhere else. Although this is a simple example, the general concept underlies the whole field of spatial analysis. Temporal dependence may be present in any type of network data, but spatial dependence is unique to spatial network data. It limits dependencies between nodes in the network that are spatially distant, thus allowing for sparse model structures.

Spatial dependence can be measured using an autocorrelation function that makes use of a spatial weight matrix. The two best known measures of spatial autocorrelation are Moran's I (Moran, 1950) and Geary's C (Geary, 1954) which are both global indices. Of these, Moran's I has been adopted most widely in the literature.

2.5.5 Spatial Heterogeneity

Spatial data often exhibit structural instability over space, which is referred to as heterogeneity. Heterogeneity has two distinct aspects; structural instability as expressed by changing functional forms or varying parameters, and heteroskedasticity that leads to error terms with non-constant variance (Anselin, 1988). Ignoring it can have serious consequences including biased parameter estimates, misleading significance levels and poor predictive power. Anselin (1988) provides some methods for testing for heterogeneity. Additionally, a number of local indicators of spatial association (LISA), a term coined by Anselin (1995), have been devised. These include a local variant of Moran's I (Anselin, 1995) and Getis and Ord's $\mathrm{G_i}$ and $\mathrm{G_i^*}$statistics (Getis and Ord, 1992), which measure the extent to which high and low

values are clustered together. A description of these and other spatial autocorrelation indicators can be found in (De Smith et al., 2007).

2.5.6 Interaction of spatial and temporal characteristics on networks

Dealing with spatio-temporal data involves dealing with the dynamics of spatial and network processes. The spatial and temporal dimensions in isolation provide a narrow view of the behaviour of a network process, but when combined they can describe the evolution of the process over time. If the spatio-temporal dependence in a network can be modelled then one essentially has predictive information. However, modelling space-time data is rarely simple as it combines the problems of dealing with spatial and temporal processes separately. The behaviour of a variable over space differs from its behaviour in time. Time has a clear ordering of past, present and future while space does not and because of this ordering isotropy has no meaning in the space-time context (Harvill, 2010). In time, measurements can only be taken on one side of the axis; hence estimation involves extrapolation rather than interpolation (Kyriakidis and Journel, 1999). Temporal data also has other characteristics, such as periodicity, that are not common in spatial data and scales of measurement also differ between space and time and are not directly comparable. Adequately accounting for these differences is essential in a model characterizing spatio-temporal behaviour (Heuvelink and Griffith, 2010).

2.6 Models of spatio-temporal network data

In this section, the main approaches to space-time modelling of spatial network data are introduced. The review covers methods that are capable of being applied to network structures. As such, many of the methods are not applied to data that would be conventionally viewed as network data, but to other types of spatial data that can be represented in a network structure. According to the scope of this thesis, the review is intentionally brief. A more detailed review of the literature on space time modelling of traffic data is given in Chapter 3. Broadly, there are two approaches that can be taken to model a space-time network process. The first approach is to explicitly model the spatio-temporal relationship between locations in a parametric framework. The second approach is to learn the dependencies form the data using a machine learning (ML) approach. These two approaches are summarised in section 2.6.1. Following this, in sections 2.6.2 and 2.6.3, the

parametric and ML models that can be applied to spatio-temporal and network data are reviewed, respectively.

2.6.1 Parametric versus machine learning methods: An overview

The literature on space-time forecasting can be broadly separated into two categories of model that come from two different modelling paradigms. The first type is the statistical parametric model. A parametric model is one where the functional form of the relationship between the dependent and independent variables is described by a set of parameters. Linear least squares regression is the archetypal parametric model. Because of their unique properties, specialised parametric techniques are typically used to model spatio-temporal data. They are usually based on some form of autoregression (AR), whereby the current value of a series is modelled as a function of its previous values. The space-time autoregressive integrated moving average (STARIMA) model is a general parametric modelling framework for spatio-temporal data (Pfeiffer and Deutsch, 1980). Such models are theoretically well founded and are able to deal explicitly with the fundamental properties of spatio-temporal data. They are incredibly powerful and inherently interpretable provided that the model assumptions hold. However, these assumptions, such as stationarity, are often difficult to fulfil (Cheng et al., 2011). As more and more real world datasets become available at higher spatial and temporal resolutions, the task of describing space-time structures with conventional methods becomes more difficult.

ML provides an alternative approach. ML models were borne out of the need to model complex, nonlinear, non-stationary relationships in massive, real world datasets, often where little is known about their functional form. They typically make minimal assumptions about the process that generates the data, and instead seek to find a function that generalises best to unseen data. A function with good generalisation ability can be seen to implicitly capture the underlying process generating the data, which is commonly referred to as learning. This implicit learning is a fundamental difference between statistical and machine learning methods and is shown graphically in figure 2.2.

Figure 2.2 – Comparison of nature, statistical models and machine learning: a) in nature, input variables x are related to response variables y by some natural process. b) Statistical models assume a stochastic data model that must be estimated. c) Machine learning assumes processes are unknown and seeks a mapping function from x to y. (Example reproduced from Breiman (2001).

The general learning task is usually to find a mapping function between a limited set of input/output pairs $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, referred to as training data, where N is the number of training examples. This task is known as supervised learning[4]. If the data are continuous then this is a regression problem, however, the approach taken to solve such problems in machine learning is different to that of classical statistics. Usually, in regression one seeks to minimise the empirical risk/error, which essentially means seeking the function that maps the inputs to the outputs with the lowest error, based on a loss function such as the sum of squared errors (SSE). This is, for example, the goal of linear least squares regression. For empirical risk minimisation to be a sensible goal, the data must be independent and identically distributed (i.i.d.) so that all future examples follow the same distribution and can be described by the same function. In reality, the data available for training a model are often noisy, and empirical risk minimisation based on a small number of training samples can lead to overfitting and poor generalisation ability. The problem is exacerbated when one is searching within a space of nonlinear functions that machine learning algorithms allow. For example, given a sufficient number of neurons, an ANN will fit the input data exactly, resulting in zero training error but poor generalisation ability. Conversely, there is also the problem of underfitting that occurs when the chosen function is too simple, which also leads to poor generalisation ability.

[4] If there are no associated outputs then it becomes a problem of unsupervised learning. This is not the focus of this thesis.

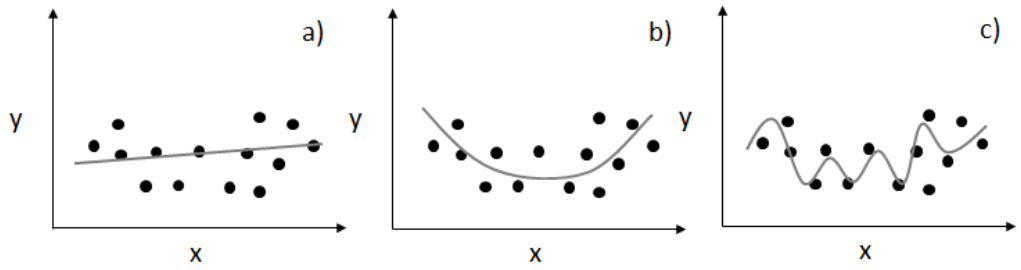


Figure 2.3- The concept of overfitting and underfitting

Finding an appropriate model can be seen as a trade-off between model bias and variance. In this context, bias is the expected generalisation error of the model, while variance is the extent to which the variance in the training data is modelled. Figure 2.3 a) represents a model with low variance but high bias, whereas figure 2.3 c) has low bias but high variance. Both models would have poor generalisation ability. Figure 2.3 b) has a better trade-off between the two and would be expected to generalise better to unseen data. One of the major contributions of the machine learning literature is the development of techniques for learning functions that attempt to control the bias variance trade off. Cross-validation is a simple example of this, whereby one seeks to find a function of the training data that is best able to predict an independent testing or validation set. k-fold cross-validation is often used whereby the training data is partitioned into k subsets which are each predicted in turn using the remaining $k-1$ subsets. In this way, the generalisation error is minimised rather than the empirical error. This type of approach has been adopted in the spatial sciences with k-fold cross validation being important in Geographically Weighted Regression (GWR) model selection (Fotheringham et al., 2002). Other types of cross-validation include leave one out, holdout and bootstrapping, which are compared by Kohavi (1995). Browne (2000) provides a detailed review.

A complementary approach to cross-validation is regularization, which involves penalising a model based on its complexity, or number of parameters. The Akaike Information Criterion (AIC, Akaike, 1974) is a method commonly used for parameter selection in statistical models that controls the tradeoff between bias and variance. ARIMA models are often fit based on the AIC. In ML, the structural risk minimisation principle is often used for model selection, which makes use of the Vapnik-Chervonenkis (VC) dimension (Vapnik and Chervonenkis, 1971) to determine probabilistic upper bounds on the training error over a set of hypothesis spaces. The hypothesis space with the tightest error bound can be chosen as the best model (Schölkopf et al., 1999). Structural risk minimisation is the key idea behind Support Vector

Machines (SVMs). Other regularised regression models include ridge regression (Tikhonov, 1943; Hoerl and Kennard, 1970), which penalises the L_2 norm of the parameter vector, and the least absolute shrinkage and selection operator (LASSO) that penalises the L_1 norm, and performs simultaneous variable selection and regression (Tibshirani, 1996).

Machine learning algorithms typically are not concerned with the physics of the processes that generate data and a successful model is one that is best able to generalise to unseen data. Consequently, they often do not have the explanatory power or interpretability of parametric models. For a spatial scientist, this may be a hindrance as typically the goal of spatial analysis is to explain spatial processes rather than predict them. Nevertheless, there has been considerable research in machine learning algorithms in the spatial sciences (Kanevski and Maignan, 2004; Kanevski et al., 2009; Kanevski, 2008). Due to their predictive capabilities, the fields of temporal and spatio-temporal analysis benefit greatly from the application of machine learning algorithms, where forecasting accuracy is valued over explanatory power.

2.6.2 Parametric Models

A range of parametric models have been applied to spatio-temporal modelling of data in a network structure. The main approaches can be separated into three catergories: statistical regression models, geostatistical models and dynamical models. These are briefly described in the following subsections.

2.6.2.1 Statistical regression Models

A range of statistical models for space-time analysis have emerged from the fusion of ideas from time series analysis, spatial analysis and spatial econometrics. Space-time autoregressive integrated moving average (STARIMA) is a family of models that extend the ARIMA time series model to space-time data (Cliff and Ord, 1975; Pfeiffer and Deutsch, 1980). STARIMA explicitly takes into account the spatial structure in the data through the use of a spatial weight matrix. The general STARIMA model expresses an observation of a spatial process as a weighted linear combination of past observations and errors lagged in both space and time. STARIMA type models have been applied to traffic forecasting (Kamarianakis and Prastacos, 2005; Wang et al., 2010), modelling coastal salinity (Saltyte, 2005), house price modelling (Beamonte et al., 2008; 2010) and forecasting of crime rates (Shoesmith, 2013) amongst others. Although the STARIMA model family accounts for spatio-

temporal autocorrelation, its parameter estimates are global. The implication of this is that the space time process must be stationary (or made stationary through differencing/transformation) for STARIMA modelling to be effective. If the space-time process is nonstationary then the residuals of model will not be uniform across time and space and the errors will not be i.i.d.

Spatial panel data models are a related set of models that have been widely applied in the econometrics literature. A panel contains observations on multiple phenomena (cross-sections) over multiple time periods, and is essentially a multivariate time series. When panel data include a spatial component they are referred to as spatial panel data, and can be seen as space-time series. Although the term *panel* describes the data itself, there are a range of models that have been developed to work with spatial panel data that originate specifically from spatial econometrics that are referred to as spatial panel data models. Introductory texts include Wooldridge (2002) and Baltagi (2005). Methodologically, they are often very similar to those encountered in the spatial statistics literature. Aspatial panel data models are modified to account for spatial dependence in one of two ways; either with a spatial autoregressive process in the error term; a spatial error model (equivalent to a spatial moving average), or with a spatially autoregressive dependent variable; a spatial lag model (Elhorst, 2003). The uptake of spatial panel data models has been much more widespread than STARIMA type models and there have been applications in liquor demand prediction (Baltagi and Li, 2006), US state tax competition (Egger et al., 2005) amongst others. In their standard form, spatial panel data models are global models and do not account for spatial heterogeneity. The autocorrelation in the system must be explained by a single autocorrelation parameter, which has the same drawbacks as STARIMA type models.

In the past 15 years, there has been a great deal of research into adapting the traditional global space-time models to deal with local characteristics of space-time data. Elhorst (2003) defines a set of spatial panel data models that account for heterogeneity. In their geographically and temporally weighted regression (G(T)WR) model, Huang et al. (2010) incorporate both the spatial and temporal dimensions into the weight matrix to account for spatial and temporal nonstationarity in GWR. Considerable research has also taken place in the traffic forecasting literature into the extension of STARIMA type models to local and

dynamic spatio-temporal network structures[5] (Min et al., 2009, 2010; Ding et al., 2010; Min and Wynter, 2011; Kamarianakis et al., 2012).

2.6.2.2 Geostatistical Models

Space-time geostatistics is concerned with deriving space-time covariance structures and semivariograms for the purpose of space-time interpolation and forecasting. The aim is to build a process that mimics some patterns of the observed spatiotemporal variability, without necessarily following the underlying governing equations (Kyriakidis and Journel, 1999). The first step usually involves separating the deterministic component $m(u,t)$ of space time coordinates u and t. Following this, a covariance structure is fitted to the residuals. The spatial and temporal dimensions can either be treated separately then combined in a sum (zonal anisotropy) or product (separable) model (Kyriakidis and Journel, 1999), or modelled jointly (Gneiting, 2002). Combinations of the two approaches have also been described in the literature (Heuvelink and Griffith, 2010). Once an appropriate space-time covariance structure has been defined, one can use standard Kriging techniques for interpolation and prediction; Bogaert (1996) provides a comparison of some of these methods.

Like global statsitcal space-time models, the success of space-time geostatistical techniques is dependent on the ability to extract a stationary space-time process from a space-time dataset. Highly nonstationary spatio-temporal relationships require a very complicated space-time covariance structure to be modelled for accurate interpolation and forecasting to be possible. Furthermore, as the main function of space-time geostatistical models is space-time interpolation, they perform less well in forecasting scenarios where extrapolation is required (Heuvelink and Griffith, 2010).

Recently, research in geostatistics has been directed towards locally varying model structures, reflecting the progress in statistical modelling of space-time data. Boisvert and Deutsch (2010) incorporate local anisotropy into geostatistical models of locally varying geological features. (Gething et al., 2007) present a local space-time Kriging model for malaria outpatient prediction. The globally stationary random function model is replaced with a locally varying one, which reduces the mean imprecision of the result. However, it does not reduce the mean bias, indicating that the local model does not fully account for the variability in the data. This highlights the difficulty of modelling the heterogeneity in spatio-

[5] These are discussed in detail in Chapter 3.

temporal data. Furthermore, the local method requires the calculation of sample space-time variograms for each location, which is computationally expensive.

2.6.2.3 Dynamical Models

Dynamical spatio-temporal models combine the spatial and temporal components of a space-time process in a model that is temporally dynamic and spatially descriptive (Wikle and Cressie, 1999). A dynamical model is built from conditional probability distributions, modelling how the current value of a space-time process evolves from its nearby past values (Cressie and Wikle, 2011). An example of such a model is the space-time state-space formulation, modelled using a space-time Kalman filter (Huang and Cressie, 1996; Wikle and Cressie, 1999; Cressie and Wikle, 2002).

A more general approach is the Bayesian hierarchical space-time model (BHSTM), which is a modelling framework designed specifically to deal with spatial and temporal variability. The hierarchical model is designed to achieve more flexible models for processes distributed in space and time. The model comprises five stages: 1) the specification of a measurement error process as a state process; 2) the specification of site-specific time series models for the state variable; 3) the estimation of a joint distribution of the time series model parameters in space using a Markov random; 4) specification of priors on the parameters, and; 5) specification of the priors of the hyperparameters. Wikle et al. (1998) apply the BHSTM to the modelling of monthly averaged maximum temperature. The approach uses temporally fixed parameters, but it is noted that time varying parameters can be accommodated in the framework.

2.6.3 Machine learning models

Despite their advantages in modelling nonlinear relations in data, machine learning models have mainly been applied to purely spatial or purely temporal processes to date. For example, ANNs have been widely applied to time series with applications in forecasting of electricity prices (Szkuta et al., 1999), electric load (Park et al., 1991; Amjady, 2001), macroeconomic variables (Swanson and White, 2011), streamflow (Zealand et al., 1999), financial time series (Kaastra and Boyd, 1996; Yao and Tan, 2000) and wind power (Kariniotakis et al., 1996) amongst many others. Frequent comparisons with conventional methods such as ARIMA have often yielded positive results. Kanevski and Maignan (2004) and Kanevski et al. (2009) describe applications of various types of ANN to spatial and

environmental modelling including radial basis function neural networks (RBFNN), general regression neural networks (GRNN), probabilistic neural networks (PNN) and neural network residual Kriging (NNRK) models to geographical problems and have gained excellent results. The authors note that the strength of ANNs is that they learn from empirical data and can be used in cases where the modelled phenomena are hidden, non-evident or not very well described. This makes them particularly useful in modelling the complex dependency structures present in space-time data that cannot be described theoretically. Hsieh (2009) also provides a good review of ANN methods applied to spatial problems. The drawback of ANN models is that their internal structure is often not interpretable, leading to the *black box* moniker.

Another widely used machine learning technique is the support vector machine (SVM). SVMs are a set of supervised learning methods originally devised for classification tasks that are based on the principles of statistical learning theory (Vapnik, 1999). SVMs make use of a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimisation theory. The key to their strong performance is that the learning task is formulated as a convex optimisation problem meaning that, for a given set of parameters, the solution is globally optimal provided one can be found. Therefore, SVMs avoid the problem of getting stuck in local minima which are traditionally associated with ANNs. This has led to SVMs outperforming most other systems in a wide variety of applications within a few years of their introduction (Cristianini and Shawe-Taylor, 2000).

SVMs were extended to regression problems with the introduction of ε-insensitive (ε)-SVR (Drucker et al., 1996). Time series forecasting was one of the application areas where it quickly demonstrated strong performance. Müller et al. (1997) were the first to apply the method, comparing its performance to RBF neural networks. Even at this early stage, the SVR algorithm outperformed the best available result for one of the test datasets used (Santa Fe competition dataset D) by 29%, showing its promise. However, the authors noted that the pre-processing step used to stationarize data is vital in the performance of the algorithm. Since then, SVR has been used to model time series in a number of application areas including financial time series (Lu et al, 2009), travel time forecasting (Wu et al, 2003; 2004), traffic flow forecasting (Castro-Neto et al, 2009), power load forecasting (Niu et al, 2009), monthly streamflow forecasting (Maity et al, 2010), rainfall forecasting (Pai and Hong, 2007) and water demand forecasting (Msiza et al, 2008).

Compared to time series analysis, the uptake of SVM in the spatial sciences was initially slow but has seen a rapid increase in popularity in the past 5 years or so, with implementations being seen in various fields such as soil classification (Bhattacharya and Solomatine, 2005) and regression (Ballabio, 2009), land cover estimation (Walton, 2008), rock depth variability (Sitharam et al., 2008) and various remote sensing applications (Mountrakis et al., 2010) amongst others. The work of Mikhail Kanevski, Alexei Pozdnoukhov and colleagues has done much to popularise the use of SVR, as well as other machine learning methods in spatial data analysis and the book "Machine Learning for Spatial Environmental Data" (Kanevski et al., 2009) provides a good introduction to some of the machine learning methods currently being used to model spatial data.

Kanevski and Canu (2000) were the first to apply SVR to spatial data mapping. They use the case study of Caesium contamination in the Briansk region of Russia to investigate the feasibility of SVR for spatial interpolation. They conclude that SVR mapping is a powerful and flexible approach that enables the user to extract information at different scales by varying the parameters over their full range. In particular, they note the potential of using large kernel bandwidths to detrend spatial data. However, parameter estimates are global for each model and (Gilardi and Bengio, 2001) demonstrate that local SVR models, i.e. one model for each regression point, produce better results than global models with the obvious drawback of having to produce a separate model for each point.

A subsequent study by (Kanevski et al., 2004) compares the performance of MLP and SVR in modelling of large scale non-linear structures in data. The results are analysed by using geostatistical models to simulate the residuals at local scales. The MLP is found to perform better than SVR in this case, however, Pozdnoukhov and Kanevski (2006) extended the SVR algorithm to deal with multiple spatial scales using multi-scale kernels (termed multiple kernel learning (MKL) in later papers). In this approach, RBF kernels of varying widths are combined linearly to model large and small scale geographical variations simultaneously, allowing the model to account for the non-stationarity in the data. This leads to the potential for much more informative prediction maps (a full description of the implementation and case study can be found in Kanevski et al. (2009).

Kanevski et al. (2007) reviewed the use of various kernel based ML methods in mapping of environmental data including probabilistic neural networks (PNN), general regression neural networks (GRNN) and multi-scale SVR (as described above). They conclude that, generally, ML methods have a wider field of application than traditional geostatistics due to their

ability to deal with multi-dimensional data. They are also well suited to dealing with large databases and long periods of observation. Additionally, they note that the SVR approach in particular is favourable because it avoids the curse of dimensionality faced by ANN methods.

2.7 Chapter Summary

In this chapter, the progress to date in modelling networks has been briefly reviewed. In accordance with the aims of the thesis, focus has been placed on spatial networks. Spatial network data combines the challenges of modelling time series and spatial series, which are autocorrelation, nonstationarity, heterogeneity and nonlinearity. The space-time models that have been applied to spatial network data attempt to account for one or more of these properties either explicitly or implicitly. The choice of method depends on the characteristics of the dataset. Modelling and forecasting of highly nonlinear, nonstationary and heterogeneous processes requires models that are able to account for these properties. In the following chapter, road networks are introduced as a type of technological network on which traffic data are collected. Traffic data are known to exhibit nonlinear and nonstationary temporal patterns as well as heterogeneity across space. The progress to date in modelling road network data is critically reviewed in its ability to model the characteristics of traffic data.

Chapter 3 – Spatio-temporal forecasting of road network traffic

3.1 Chapter overview

The task of traffic forecasting using real traffic data involves leveraging knowledge of the past and current conditions on the road network to make informed forecasts of future conditions. This can be accomplished in a wide variety of ways that vary in their sophistication from simple historical averaging to complicated neural network models. It is only in the latter part of the 20th century that large scale, urban network based traffic datasets have become available and the power of computers has become sufficient to make analysis on a network scale possible. Hence, forecasting of traffic data has traditionally been achieved using univariate time series methods, where data collected at a single location are used to predict future conditions at the same location in an autoregressive framework. The simplest univariate techniques include historical averaging and smoothing methods (Williams et al., 1998). More sophisticated approaches exist that can be broadly separated into two categories; parametric methods and machine learning methods. Parametric methods assume that the functional form of the relationship between dependent and independent variables is known and can be described by model parameters. Machine learning methods such as nonparametric regression and neural networks make no such prior assumption. They are data driven methods that "learn" the characteristics of the data as the dataset grows (Vlahogianni et al., 2004).

In this chapter, the state of the art in space-time forecasting of traffic variables is reviewed. The aims of this chapter are threefold: 1) To examine the ways in which spatio-temporal information is incorporated into space-time forecasting models; 2) To examine how missing data is dealt with in forecasting models, and 3) To identify the main challenges in space-time forecasting of traffic data that should be addressed in this thesis. The chapter begins with an introduction to road networks in section 3.2, followed by a description of the types of traffic data that are collected from the traffic stream in section 3.3, and a brief treatment of the nature of urban traffic and the causes of urban traffic congestion in section 3.4. Following this, in section 3.5, a general review of parametric and machine learning methods is given in order to highlight the relative strengths and weaknesses of the two approaches. In section 3.6, a more in depth review is given of the techniques that have been used to date to incorporate spatial information into traffic forecasting models. Then, in section 3.7, the methods that have been used to deal with missing data in traffic series are discussed. Finally,

in section 3.8, the achievements and challenges of space-time forecasting of traffic data are summarized.

3.2 Road Networks

The road network is among the most recognisable types of network, and is an important part of the daily lives of human beings. The efficient and safe operation of road networks is vitally important economically, environmentally and socially. For this reason, road networks have been highly studied for decades. The study of road network structure predates the field of complex networks and has long been of interest due to its inherent impact on transportation system performance (Xie and Levinson, 2007). Garrison (1960) explored the potential for graph-theoretic research into transportation systems and computed a range of structural measures on a subset of the US interstate system including connectivity, diameter, central place and dispersion. Other early studies followed similar approaches (Garrison and Marble, 1962) constrained by the lack of data availability and computational power at the time. The representation of transportation networks as graphs is pervasive in the literature, and most of the traffic simulation models in use today use a graphical structure for traffic assignment.

3.2.1 Graphical representation

Road networks are usually represented as planar networks. In this representation, each intersection is a node and the road segments between intersections are edges. Road networks can be seen as an example of a random network, which entails a random degree distribution (Xie and Levinson, 2007). Due to their strong geographical constraints, road networks exhibit fractal scaling, meaning that they are geometrically self-similar (Shen, 1997). Although physically planar, road networks need not be represented as such. The topology of a road network can be based on street-street intersection rather than road segment intersection. Using this representation, Jiang (2007) observed that the topologies of urban street networks exhibit scale free properties and follow a power law distribution, with 80% of streets having length or degree less than average and 20% having length or degree greater than average. This property has subsequently been shown to apply to traffic flow on street networks (Jiang, 2009). The findings of these studies support the intuitive understanding that the road network is a hierarchical system, which is reflected in the real life ranking of road types (i.e. M, A, B etc. in the UK). One uses the so called ‘main roads’ to travel longer distances while using smaller roads either to travel locally or to access the main

roads. Road networks have also been shown to exhibit the small world phenomenon. Xu and Sui (2007) used autocorrelation analysis to show that the small world phenomenon on transportation networks is due to the interplay between network structure and the dynamics taking place on the network.

3.3 Traffic Data and sensor networks

Nowadays, the road networks of major cities are equipped with sensor networks that measure traffic properties such as speed (v), flow (q), density (k) and travel time (TT) directly from the traffic stream (see table 3.1). This means that a wide range of space-time series pertaining to traffic conditions may be available, depending on the infrastructure installed in a particular city. Traffic sensors can be either static or dynamic. Static sensors, such as loop detectors or automatic number plate recognition (ANPR) cameras measure traffic conditions at a fixed point, or along a section of road called a link. Dynamic sensors are those positioned on vehicles moving within the traffic stream, such as GPS devices and mobile phones (Hofleitner et al., 2012; Kriegel et al., 2012). Such vehicles are referred to as floating vehicles. This review focusses only on static sensors.

3.4 The nature of urban road traffic

In its early stages, the development of traffic theory and methodology was focused mainly on highways, where traffic flows are uninterrupted and there are relatively few entry and exit points to the traffic stream. On highways, the propagation of traffic conditions along the traffic stream can be described by macroscopic flow theory, which states that traffic streams behave similarly to fluid streams. Lighthill and Whitham (1955) used the method of kinematic waves to deduce a theory for the propagation of changes in traffic distribution along long crowded roads based on flood movements in rivers. In a free flowing situation traffic flows from upstream to downstream, so vehicles observed at one location at one time point will be observed at another location downstream at a later time point. However, regions of increased traffic density move slightly slower than the mean vehicle speed and vehicles will experience a sudden decrease in speed on joining them. This phenomenon can be observed as a shockwave that propagates in the opposite direction to the traffic stream. Richards (1956) further developed the theory of shockwaves and the collected contribution of the three authors became known as the Lighthill, Witham, Richards (LWR) model.

Table 3.1 – Description of traffic variables

Property	Definition	Measurement
Flow (q)	The rate at which vehicles pass a fixed point during a given period.	e.g. Vehicles per hour.
Density/occupancy/ Concentration (k)	The number of vehicles (N) on a roadway (L).	$k = \frac{N}{L}$
Speed (v)	Distance travelled per unit of time.	Time mean speed: $v_t = \frac{1}{m}\sum_{i=1}^{m} v_i$ Where m is the number of vehicles passing a fixed point, e.g. a loop detector. Space mean speed: $v_s = (n/\sum_{i=1}^{n}(\frac{1}{v_i}))$ Where n is the number of vehicles passing the roadway segment. Usually recorded using pictures or video.
Travel time (tt)	The time taken to travel between two fixed points.	Individual vehicle travel time TT: $TT = t_2 - t_1$ Where: t_1 = the time the vehicle began its journey t_2 = the time the vehicle completed its journey Aggregated travel time of road link l at time period p: $TT_l(p) = 1/N\sum_{i=1}^{N} TT_{l,i}(t^*), t^* \in p$ Where: N = the total number of vehicles traversing the link in time period p. Source: (Liu, 2008)

Traffic on urban networks is fundamentally different to traffic on highways, primarily because it is interrupted by traffic signals at intersections. Aside from this, there are various other factors that determine the level of traffic on urban networks, which are discussed briefly below.

3.4.1 Supply and demand relationship

The underlying factor that determines the speed of the traffic stream on both highways and urban roads is interaction between the demand for road space and the supply of road space, known as capacity (Bell and Iida, 1997). The relationship can be shown graphically using one of the fundamental tools in transportation engineering, the macroscopic fundamental diagram, shown in figure 3.1. On the diagram, when demand is lower than supply, conditions are said to be *free flowing* (S_f). When density reaches a critical density (D_0), the maximum flow (V_m) is reached and demand and supply are equal. As density continues to increase past this critical level demand outstrips supply, flow decreases towards the jam density (D_j) and the road is said to be *congested*. Congestion is characterised by slow moving stop and go traffic and is very frustrating and costly for road users.

Source: Gordon and Tighe (2005, p. 167)

Figure 3.1 – The macroscopic fundamental diagram

Congestion that results purely from the level of demand outstripping the capacity of the road way is referred to as *recurrent congestion*. Recurrent congestion occurs on a daily basis,

and is characterised by rush hour or peak traffic, determined by commuting patterns. In the peak periods, the mean level of traffic is higher. Traffic is also more variable in the peak periods, resulting in heteroskedasticity (Anacleto Junior et al., 2012). The level of demand varies between weekdays and weekends, resulting in different traffic patterns on different days of the week. The distribution of urban traffic variables has received attention recently. Stathopoulos and Karlaftis (2001) show that traffic flows are not normally distributed, and that the distribution varies with time of day, day of the week and between seasons. Fosgerau and Fukuda (2012) examine the distribution of travel times and demonstrate that the median and interquartile range are higher in peak periods. The time varying characteristics of traffic data has led to attempts to characterise traffic into distinct regimes, such as free flowing and congested (Kamarianakis et al., 2010).

3.4.2 Road type

Urban roads can be classified hierarchically into different types based on their function. The top of the road hierarchy is the motorway/highway, coded with the letter M in the UK. This type of road is common in cities in the US, but less so in the UK and Europe. The major motorway serving London is the M25, which encircles the city centre at a radial distance of between 12.5 and 19.5 miles of Charing Cross (historically used as the centre of London). Below this, arterial roads are major roads that carry traffic between locations within the city, coded with the letter A. Arterial roads are connected to localities via collector roads, which are generally coded with the letter B. Localities are served by local roads, which do not have a letter classification, and are referred to by their street names. A radial road is a particular type of road that connects suburban areas with the city centre. Radial roads are generally motorways or arterial roads.

Arterial roads generally exhibit heavier traffic, higher speed limits, and higher capactiy than collector roads, with local roads experiencing the lowest levels of traffic and lowest speed limits. The nature of the peaked profile of each of these types of road depends on their direction. For instance, radial roads can be categorised into inbound and outbound routes. Inbound routes tend to have higher traffic in the morning peak period than the afternoon peak as they deliver commuters to work in the city centre, and vice versa. This affects the distribution of their traffic (Stathopoulos and Karlaftis, 2001). Roads within the city centre tend to be more congested throughout the day as they are in constant use.

Table 3.2 – Delays caused by traffic signals

Type	Description
Stop, acceleration and deceleration	Delays incurred when decelerating before a red signal, waiting at the red signal and accelerating away from the red signal.
Vehicles in platoon	Delays incurred when the arrival rate during the red phase outstrips the exit rate during the green phase. This causes some vehicles to wait for two red phases, thus increasing their travel time.
Signal control strategy	Delays caused by unsuitable signal control strategy. For example, fixed signal control strategies are only suitable for the traffic conditions they were designed for, and can cause delays when conditions differ. In London, a large proportion of the traffic signals are dynamically controlled by split cycle offset optimisation technique (SCOOT), which is an adaptive system that responds automatically to fluctuations in traffic flow through the use of on-street detectors embedded in the road (Imtech, 2013).
Signal offsets	Routes consist of several signalised intersections in a row. If the green phases of these intersections are correctly offset then they can minimize interruptions in flow in the traffic platoon. However, incorrectly offset signals can cause increased delays.
Queue overflow	When the queue at a signal fails to clear during the green phase, a residual overflow queue remains. This occurs when the demand exceeds the capacity of the intersection. If this situation persists for multiple signal cycles then the overflow queue continues to grow causing overflow delays. In extreme cases, a queue may occupy an entire road section, thus blocking exits from the upstream intersection. This causes congestion to propagate from downstream to upstream.

3.4.3 Signalisation

Traffic signals are the main tool that traffic controllers use to regulate traffic in urban areas. Traffic signals are used both to control the flow of road traffic, and to allow pedestrians to safely cross roads. They can have a fixed cycle, or they can be controlled either manually or automatically[6]. The pattern of recurrent delays on urban roads is determined to a large extent by signalized intersections. These delays can be separated into 5 categories, which are: 1) delays of stop, acceleration and deceleration; 2) delays of vehicles in platoon; 3) delays of signal control strategy; 4) delays caused by signal offsets and 4) delays of overflow queues. These are summarized in table 3.2, more details can be found in (Liu, 2008, pp.16-22).

Traffic signals are responsible for a portion of the stochastic variation present in observations of traffic variables. The extent to which traffic signals affect observations depends on the temporal resolution of the data. In data of high temporal resolution, it is possible for the red and green phases to align with the data collection interval, inducing variability that does not reflect the true traffic state. Lower resolution data smooths this variation.

3.4.4 Non-recurrent events

Demand for road space, road type and signalisation determine the *recurrent* traffic patterns on urban roads. There is considerable variation around these patterns, which can largely be attributed to non-recurrent events (Skabardonis et al., 2003). Events can be either planned or unplanned. The effects of planned events, such as scheduled roadworks, tube strikes and special events, can be mitigated to an extent by careful planning on behalf of traffic engineers. Unplanned events, such as emergency roadworks, incidents, demonstrations, inclement weather and burst water mains, cannot be anticipated and their effects can be severe (Kwon et al., 2006).

Non-recurrent events either decrease the capacity or increase the demand on the roadway temporarily. In these cases, congestion occurs that would not usually be present. This is referred to as *non-recurrent* congestion. Non-recurrent events account for the heavy tail in the distribution of traffic (Börjesson et al., 2012). The effect of non-recurrent events is exacerbated during the peak periods when demand is higher because the network is nearer

[6] An overview of the different types of signal control and a description of signal control variables are given in appendices A.1 and A.2 repsectively.

to saturation, making it highly sensitive to disruptions. This leads to increased variance (or interquartile range) in traffic data during the peaks (Fosgerau and Fukuda, 2012), which is termed heteroskedasticity in the statistical literature (see section 2.5.1.4). Non-recurrent congestion is inherently unpredictable, but the ability to forecast its effects is of critical importance (Vlahogianni, 2007). Despite this, little attention is given to performance during non-recurrent congestion when evaluating the performance of forecasting models (Vlahogianni et al., 2004).

3.4.5 Other factors

Cities are complex, multifunctional systems with many actors competing for finite resources and space. The traffic stream is a competitive environment where cars share space with other forms of transportation such as buses, heavy goods vehicles (HGVs), bicycles and emergency vehicles. Vlahogianni et al. (2006) shows that the speed-volume relationship on urban arterials is affected by traffic composition. Motorcycles, taxis and HGVs are all shown to be critical in determining the relationship.

3.5 Spatio-temporal forecasting of traffic data

Because of the many factors that determine urban traffic, only a small proportion of which are observed, it is very difficult to model traffic as a physical process. Therefore, the majority of the literature on traffic forecasting focuses on data-driven approaches. In such approaches, statistical relationships are extracted from historical data, and used to forecast future traffic conditions using information about the current traffic conditions. In this section, the literature on data driven forecasting of traffic data is reviewed. In accordance with the review in Chapter 2.6, the literature is divided into the two main categories of forecasting model: parametric models and machine learning models. They are reviewed in turn in sections 3.5.1 and 3.5.2. Traffic simulation models are briefly introduced in section 3.5.3, but are not mentioned in the remainder of the review as they are not the focus of this dissertation. Following this, in section 3.5.4, the strengths and weaknesses of the two approaches are summarised.

3.5.1 Parametric Methods in traffic forecasting

Parametric statistical models are widely used in the traffic foreacasting literature and the method that has received the most attention is the ARIMA model family (Box et al., 1994). ARIMA is a popular statistical time series forecasting method that has been applied in a wide

range of fields. ARIMA was first used for traffic forecasting as early as the late 1970s (Ahmed and Cook, 1979; Levin and Tsao, 1980) and has been used in numerous studies since (Hamed et al., 1995; Lee and Fambro, 1999; Billings and Yang, 2006; Guo and Smith, 2007, amongst others). To account for the inherent seasonality in traffic data, seasonal ARIMA (SARIMA) models can be used, which can perform better than standard ARIMA models (Williams et al., 1998; Smith et al., 2002; Williams and Hoel, 2003; Guin, 2006). Generalised autoregressive conditional heteroskedasticity (GARCH) models have also been used for traffic forecasting (Kamarianakis et al., 2005). The GARCH model allows the conditional variance of the series to be computed, allowing for forecasting confidence interval construction which adds interpretability to the model output. The SARIMA and GARCH models have also been combined in composite models (Guo, 2005, Guo and Smith, 2007). Statistical models have been adapted to deal with space-time data. Most examples have been developed based on the ARIMA model family (Box et al., 1994) including multivariate ARIMA models, vector autoregressive (VARMA) models and space time autoregressive integrated moving average (STARIMA) models and variants (Pfeifer and Deutsch, 1980). Although all of these can operate on space-time data, STARIMA is the only one that explicitly accounts for spatial structure in the data through the use of a spatial weight matrix.

A large proportion of the literature has focussed on the application of state-space models to traffic forecasting (Okutani and Stephanedes, 1984; Whittaker et al., 1997; Stathopoulos and Karlaftis, 2003; Liu et al., 2006a; Xie et al., 2007; van Lint, 2008). State space representation is a mathematical model of a physical process as a set of input, output and state variables. In traffic prediction, Kalman filtering (Kalman, 1960) has been widely used to solve state space models.[7] A Kalman filter takes measurements over time of a process corrupted by noise and produces values that tend to be closer to the true values of measurements than the observed values. This is achieved by computing a weighted average of a forecast of the system's state at the next time point and the corresponding new observation. Higher weight is given to the value with the least uncertainty. State-space models are very similar to autoregressive moving average with exogenous variables (ARMAX) models but have a number of advantages; the components of the series are modelled separately, they can adapt to changes in the system over time, they are easily extended to the multivariate case,

[7] Although the terms *Kalman filter* and *state-space model* are often used interchangeably, they are different. The Kalman filters is a method for solving a state space model and can be used to predict other model types such as neural networks. Likewise, state space models can be solved using other methods.

they can deal easily with missing data, large models can be handled efficiently and predictions are straightforward (Stathopoulos and Karlaftis, 2003).

However, simple Kalman filters are not applicable to modelling of non-stationary time series as they require a constant mean and covariance. The extended Kalman filter (EKF) is a nonlinear extension of the Kalman filter that linearizes the current mean and covariance by replacing the state transition and observation matrices with Jacobian matrices, and has been applied to traffic forecasting. Liu et al. (2006) present two different ways of using an EKF for arterial travel time prediction. The first is a data-driven approach that uses the EKF to train a feed-forward neural network (FFNN). The second is a state space approach that uses information about free flow travel times and delays, and queuing behaviour to produce a model of travel time evolution. The models are tested using two scenarios; one where the data has been cleaned and one where it remains in its raw form. It is found that the state space EKF slightly outperforms the FFNN EKF on the clean data as the state space EKF is able to model the underlying properties of the traffic stream. However, the FFNN EKF is significantly more robust to the noise present in the raw data.

An approach that has gained popularity in a number of fields recently is the Bayesian network (also referred to as belief network and directed acyclic graphical (DAG) model). The idea of Bayesian networks is to represent the conditional independence structure of the process under study as a DAG comprising nodes and edges. It is assumed that each node in the DAG (corresponding to a variable in the model) is conditionally dependent only on its parent nodes so all of the calculations are performed locally. This makes the computations simple, even for large networks. The graphical structure also means that nodes can be added or removed simply as the network changes. Bayesian networks have been applied to traffic flow forecasting both on highways (Queen et al., 2007; Queen and Albers, 2008; Fei et al., 2011) and urban networks (Sun et al., 2004;(S. Sun et al., 2005); Yu and Cho, 2008; Dong and Pentland, 2009). They have also been applied to travel time forecasting on highways and urban networks.

3.5.2 Machine Learning Methods in traffic forecasting

The term *machine learning* encompasses a wide range of computational techniques. Some of the techniques that have been used in the traffic forecasting literature are reviewed in the following subsections.

3.5.2.1 Nonparametric regression

One of the simplest machine learning techniques that has been applied to traffic forecasting is k-nearest neighbours regression (k-NN) (Smith and Demetsky, 1996), which the authors refer to as nonparametric regression[8]. This approach involves comparing the current state vector of the system with its k nearest neighbours in the historical dataset, and producing a forecast as an average of the k neighbours. Although simple, the method performs strongly, outperforming ARIMA, neural networks and historical based methods on freeway data (Smith and Demetsky, 1997). The method does not directly account for temporal dependency but the implicit assumption is made that more similar traffic patterns will be more likely to evolve in similar ways. Nearest neighbour approaches often prove strong predictors but their main drawback is that they can be computationally very intensive. Each forecast is made by computing the distance (usually Euclidean) between the current data pattern and an historical database of patterns. There is a trade-off between richness of the historical database and the computation time required to produce a forecast.

A closely related non-parametric regression method is kernel regression (Nadaraya, 1964; Watson, 1964). Kernel regression is very similar to k-NN regression and often produces similar results (Haworth and Cheng, 2012); the difference being that the prediction output is computed as a weighted average of all the training data based on the similarity of (or distance between) the current state vector and the training set. A kernel function is used to compute the weights, which is usually a Gaussian function or some other Radial Basis Function (RBF) kernel. Although originally formulated for point data, kernel regression can be easily extended to multidimensional data and thus is suited to time series applications. Kernel regression has recently been applied to traffic forecasting (Sun and Chen, 2008; Blandin et al., 2009; Han et al., 2010; Huang et al., 2011; Leng et al., 2013) with promising results. Kernel regression has also appeared in the literature under different names, including general regression neural network (GRNN) (Specht, 1991).

Non-parametric regression techniques have a number of benefits; firstly, they are theoretically simple and can be easily understood and implemented by practitioners. Secondly, they do not have a lengthy training process that is required by statistical models and neural network models. However, they rely on all possible situations having been observed in the training set and as such may encounter problems in non-recurrent

[8] The distinction is made between k-NN and nonparametric regression here as nonparametric regression is a broad term that encompasses a wide range of regression techniques.

conditions. Additionally, the size of the historical database, the embedding dimension and the time lag must be decided and, in kernel regression, the kernel parameter(s) must be optimised.

3.5.2.2 Kernel methods

A more recent trend in machine learning has seen the application of *kernel methods* (KMs) to a wide range of problems. KMs are related to the nonparametric regression models mentioned above, but they use kernels in a different way. KMs make use of the *kernel trick* to map the input data into a high dimensional feature space, where a linear algorithm is used to find a solution that becomes nonlinear in the input space. The best known kernel method for regression is support vector regression (SVR), which is based on the principles of statistical learning theory (Vapnik, 1999). SVR has three main strengths; firstly, it uses structural risk minimisation rather than empirical risk minimisation so it has strong generalisation ability; secondly, through the use of kernels it can be converted into a non-linear algorithm; and thirdly, its solution depends only on a subset of training examples known as the support vectors, so its solution is sparse. Kernel ridge regression (KRR) is a related method based on linear least squares regression (Saunders et al., 1998; Shawe-Taylor and Cristianini, 2004). KRR is simpler to implement than SVR but does not have the advantage of sparseness. KRR with certain kernel functions is referred to as Gaussian processes regression (GPR) (Williams and Rasmussen, 1996; Rasmussen and Williams, 2006) and Kriging (Krige, 1951).

Wu et al. (2004) first used SVR for the forecasting of travel times on Taipei's freeway system. Travel times over three distances for 28 consecutive days are used to train a model to forecast the following 7 days data in a one-step-ahead scenario. The results are compared to a current-time predictor and a historical mean predictor and are found to be superior in all cases. Vanajakshi and Rilett (2004; 2007) compare the performance of SVR and ANNs in forecasting travel times on San Antonio's freeway system using a forecast horizon varying from 2 minutes to 1 hour ahead. They find that SVR performs better than ANN when the size of the training set is small. The SVR solution is dependent only on the support vectors and is therefore more robust and has greater generalization ability.

It is possible to train an SVR model in an online or sequential manner. Castro-Neto et al. (2009) took advantage of this fact by implementing an online SVR algorithm for short term traffic flow forecasting. Instead of retraining the model every time new data become

available, which is computationally expensive, the model is iteratively updated three samples at a time and the solution support vectors are changed accordingly. The advantage of this approach is that new information is incorporated into the existing structure of the solution. Empirical results show that the method outperforms Holt's exponential smoothing, and multi-layer perceptron (MLP) ANNs. Gaussian Maximum Likelihood (GML) produces better results under typical conditions but the OL-SVR model performs well under non-recurrent conditions due to the fact it adapts well to new data. Given that non-recurrent events are more difficult to predict this is a significant benefit.

Zhang and Xie (2008) used v-SVR for highway traffic volume forecasting, and found the method to outperform a multi-layer feedforward neural network (MLFNN). In a subsequent study, Xie et al. (2010) apply GPR to highway traffic volume forecasting, and compare the results with the v-SVR model of Zhang and Xie. The results are found to be comparable, but the GPR model has the advantage of providing error bounds on the forecasts.

3.5.2.3 Artificial neural networks

ANNs are one of the most widely used and successful methods in the traffic forecasting literature, with examples dating back as far as the early 1990s. Dougherty (1995) provides a comprehensive review of the early work. ANNs are a family of non-parametric methods for function approximation that have been shown to be very powerful tools in many applications that involve dealing with complex real world sensor data (Mitchell, 1997). They are inspired by the observation that biological learning is governed by a complex set of interconnected neurons. The key concept is that, although individual neurons may be simple in structure, their interconnections allow them to perform complex tasks such as pattern recognition, classification and regression (Larose, 2004). The basic structure of an ANN is a layered, feed-forward, completely connected network of artificial neurons, termed nodes. It usually consists of three layers; an input layer, a hidden layer and an output layer. Each of the connections between nodes is weighted. The hidden layer can contain any number of nodes, which is specified by the user. Generally speaking, the more nodes in the hidden layer, the greater the ability of the ANN to identify complex patterns, however, too many nodes can lead to overfitting. Depending on the implementation, there is often no intuitive way to select the appropriate number of nodes or interpret their structure and as such they are often referred to as *black boxes*.

Although initially performance was in traffic forecasting was questionable (Kirby et al., 1997), subsequent improvements in data selection, network architecture and model training have yielded results that are at least comparable with competing methods in most cases. Because of their flexible design, incorporating spatio-temporal structure into ANNs is relatively straightforward and has been investigated in a number of studies (e.g. van Lint et al., 2002). Considerable research has been undertaken into developing ANNs for traffic forecasting. The more sophisticated approaches have been shown to produce very strong forecasting of traffic variables on highways (van Lint et al., 2002; 2005; van Lint, 2006; Chan et al., 2012a; 2012b; Chan et al., 2013a; Chan et al., 2013b) and urban networks (e.g. Vlahogianni et al., 2005; 2007). However, the basic models still suffer from lack of interpretability, meaning that their uptake by non-experts is difficult.

3.5.3 Traffic Simulation

An important modelling approach that warrants consideration in this review is traffic simulation. Traffic simulation models the propagation of traffic around the road network based on theories of the movement of traffic. It is widely used by transport planners for various aspects of traffic management, including forecasting. As this thesis focusses on data-driven approaches, simulation models will only be given a brief treatment in this section and are not included in the remainder of the review. The literature on traffic simulation can be separated into three types of model: macroscopic, microscopic and mesoscopic, each of which attempts to model traffic at a different level of detail. These are discussed in turn below. More attention is paid to macroscopic models as they are more often used for forecasting in practice.

3.5.3.1 Macroscopic simulation

Macroscopic traffic flow models (MTFMs) simulate the flow of vehicles around the network at the aggregate level, dealing with traffic flow in terms of variables such as flows, speeds and densities. They are typically based on analogies with theoretically well founded physical models of flow in continuous media, in which case they are referred to as continuum MTFMs (Hoogendoorn and Bovy, 2001). The three most commonly used models are LWR based models, Payne-type models (Payne, 1971) and Helbing-type models (Helbing, 1996). LWR models are based on kinematic wave theory. They are first order models, which means that they consider the mean speed of the traffic stream solely as a function of the traffic density, and require a single equation to describe the evolution of the traffic condition (Kotsialos and

Papageorgiou, 2001). This makes them relatively simple to implement. Payne-type models are second order models, which consider the mean speed as an independent variable and include an additional equation to describe speed dynamics. Helbing-type models are also second order models which are based on gas kinetic equations, extending Payne-type models to include an equation for velocity variance (Helbing et al., 2002).

In practice, MTFMs are usually applied by discretizing the continuous model equations in order to approximate the solution for small road segments and times. An important recent development in this regard is the cell transmission model (CTM) (Daganzo, 1994; Daganzo, 1995). The CTM divides the roadway into cells, between which traffic can flow based on certain rules, consistent with LWR theory. An example of a practically implemented second order MTFM is METANET (Kotsialos et al., 2002). Other MTFMs employed in practice include Visum (PTV Group, 2014) and Aimsun (TSS, 2014), and Kotusevski and Hawick (2009) provide a useful review of some of the alternatives.

In order to be effective, MTFMs require specification of initial and boundary conditions that describe the flows into and out of the network to be modelled. These boundary conditions should vary with time according to temporal variations in traffic levels. Currently, the main application area for MTFMs is on highway networks, where traffic is largely uninterrupted and good quality data are available from fixed loop detectors in the road surface. It is more difficult to implement MTFMs in the urban setting because urban traffic is interrupted by signals, and there often is insufficient coverage of sensors (Liu, 2008). MTFMs require good quality data relating to at least two variables from flow, speed and density[9]. It is also more complicated to specify boundary conditions in urban areas, because good estimates of demand (origins and destinations) are difficult to obtain for urban networks. However, recent research has focussed on eliciting macroscopic fundamental diagrams from large urban areas (Geroliminis and Daganzo, 2008) and extending the CTM to urban networks (Hu et al., 2010; Long et al., 2011), which will help in this regard.

3.5.3.2 Microscopic and mesoscopic simulation

Microscopic simulation models take the view that the aggregate traffic conditions observed on the network are a direct result of the heterogeneity in the movement/behaviour of individual vehicles/drivers. Two classes of such models that have traditionally been used are car following models, which attempt to model how vehicles follow one another, accounting

[9] Only travel time data are available within this project.

for reaction times and driver preferences (Brackstone and McDonald, 1999); and lane changing models, which model how vehicles overtake and prepare for turns (Barceló, 2010). Such models tend to be too computationally intensive for use on a network scale, and are better suited to the examination of scenarios such as road layout changes. More recent developments have employed cellular automata (Nagel and Schreckenberg, 1992) and agent based models (Cetin et al., 2003) to traffic simulation. Such models are more suited to real time application and allow the representation of complex behaviours in a simple and customisable way, and as such represent a major growth area in the subject.

Microscopic simulation software is provided by Aimsun (Barceló and Casas, 2005) and VISSIM (Fellendorf and Vortisch, 2010) amongst others, and Jones et al. (2004) review a number of implementations, including CORSIM, SimTraffic along with Aimsun.

Mesoscopic simulation models provide a middle ground between micro and macroscopic models. They attempt to model the behaviour of drivers without explicitly distinguishing their time-space behaviour (Hoogendoorn and Bovy, 2001). This leads to computation savings over microscopic models, whilst retaining most of the detail.

3.5.4 Comparison of methods reported in the literature

There is a burgeoning literature on traffic forecasting using both parametric and non-parametric approaches. However, there is lack of consensus over which type of model is the most suitable. Perhaps the main reason for this is that as researchers create increasingly sophisticated models it becomes more difficult to directly compare them due to the difficulty and time involved in their implementation. In general, model performance tends to be compared to the basic implementations of other models. For instance, sophisticated ANN models are often compared to simple ARIMA models, the deficiencies of which have been well documented and addressed in the literature. Additionally, the datasets used in different studies have hugely different characteristics. They may have different spatial and temporal resolutions, different noise levels, they may be highway or urban arterial, flow data or travel time data and so on and so forth. Add to this the diverse range of error indices used and direct comparisons become very difficult. Table 3.3 gives a summary of the traffic forecasting literature that has been surveyed in this section. Only those models that are compared with other models are included in the table as their accuracy relative to other methods is demonstrable. Furthermore, the traffic simulation models introduced in section 3.5.3 are not reviewed as they are not applicable in the current setting. To aid the

comparison, 8 criteria are reported aside from the method, author and publication date, which are:

1 The models with which the approach is compared
2 The aggregation level of the data
3 The forecast horizon(s) that is(are) evaluated
4 The traffic variable(s) that is(are) evaluated
5 The application area (either urban or highway)
6 The size of the network (number of locations)
7 Whether spatial information is included in the model (yes/no)
8 Whether missing data are accounted for directly in the model (yes/no)

The fields in the table are partly based on a similar comparison in Vlahogianni et al. (2004).

Table 3.3 - A comparison of forecasting models in the literature

Method	Author(s)	Date	Comparison Models	Aggregation	Horizon (mins)	Variable	AA	N	S	M
Nearest neighbour non-parametric regression	Smith and Demetsky	1997	Historical Average, ARIMA, BPNN	15 min	15min; 1,2,3,4 hours.	Flow	H	2	N	N
SARIMA	Williams et al	1998	Nearest neighbour non-parametric regression, BPNN, historical average, random walk	15min/15 min hourly flow	15 mins	Flow	H	2	N	N
ARIMA	Ahmed and Cook	1979	Double exponential smoothing, moving-average, Trigg and Leach adaptive models	-	-	Flow, Occupancy	H	166	N	N
subset ARIMA	Lee and Fambro	1999	ARIMA	5 min	5 min	Flow	H	2	N	N
Non-parametric regression (various techniques)	Smith et al	2002	SARIMA (SARIMA has greater accuracy), naïve	15 min	15 min	Flow	H	4	N	N
SARIMA	Williams and Hoel	2003	Random walk, Historical Average, deviation from historical average	15min/15 min hourly flow	15 mins	Flow	H	2	N	N
STARIMA	Kamarianakis and Prastacos	2005	ARIMA (similar performance)	7.5 min	7.5 min	Flow	U	25	Y	N
dynamic STARIMA	Min et al	2009	Chaos theory, MARS	5 min	5 min	Flow	U	50	Y	N
STARIMA	Ding et al	2011	ARIMA	5 min	5, 10, 15, 30 min	Flow	H	13	Y	N
VARMA	Chandra and Al-Deek	2008	ARIMA	5 min	5 min	Speed	H	1	Y	N
GSTARIMA	Min et al.	2010	STARIMA	5 min	5 min	Flow	U	50	Y	N
STM	Ghosh et al	2009	SARIMA	15 min	15 min, 1 hour, 12.5 hours	Flow	U	10	Y	N
Mahalanobis Kernel Regression	Sun and Chen	2008	Kernel Regression	15 min	15 min	Flow	U	5	N	N
State space (Kalman Filter)	Okutani and Stephanedes	1984	UTCS-2	15 min counts at 5 min intervals	5, 15, 30, 45	Flow	H	1	Y	N

State space (Kalman Filter)	Whittaker et al	1997	Naïve	Unclear	Up to 5 step – empircal errors not reported	Flow, density	H	10	Y	N
State space (ARMAX)	Stathopoulos and Karlaftis	2003	ARIMA	3 min	3 min	Flow	U	5	Y	N
State space (EKF)	Liu et al	2006	FFNN EKF (state space EKF better on clean data)	2 min	2 min	Travel time	U	1	Y	N
State space (Kalman Filter - Wavelet)	Xie et al	2007	State space (Kalman Filter)	5 min	5 min	Flow	H	4	N	N
BPNN	Kirby et al	1997	ATHENA, ARIMA (Both perform better)	30 min	30, 60, 120 min	Flow	H	4	N	N
TDNN with GA/CATNN	Abdulhai et al	2002	MLF NN	30 seconds	30 seconds, 1,2,4,5,15 mins	Flow	H	16	Y	N
SSNN	van Lint et al	2002	Network Level Travel Time (naïve)	1 min	1 min	Travel time (simulated) (input flow and speed)	H	12	Y	N
SSNN (missing data)	van Lint et al	2005	SSNN	1 min	1 min	Travel time (input flow and speed)	H	19	Y	Y
ensemble SSNN	van Lint et al	2006	Instantaneous predictor (naïve)	1 min	1 min	Travel time (input flow and speed)	H	27	Y	Y
state space (online censored EFK)	van Lint	2008	Naïve, Historical Average, online-delayed EKF, SSNN (inferior performance)	1 min	1 min	Travel time (input flow and speed)	H	14	Y	N
MLP, Modular Network, hybrid PCA network, CANFIS	Ishak and Alecsandru	2004	Naïve, heuristic (CANFIS generally best network)	5 min	5,10,15,20 min	Speed	H	4	Y	N
Optimized and meta-optimized neural networks	Vlahogianni et al.	2005	ARIMA (univariate model), State-space (multivariate model)	3 min	3,6,9,12,15 min	Flow	U	1	Y	N

genetically optimised modular TDNN	Vlahogianni et al	2007	ARIMA, state-space, static MLP, Genetically optimised MLP	3 min	3 min	Flow	U	1	Y	N
HPNN	Hu et al	2008	BPNN, univariate PNN, upstream/downstream PNN	5 min	5 min	Flow	H	1	Y	N
FIFO filter MLFN-ES	Srinivasan et al	2009	ARIMA, ARIMA (clustering), ANN, ANN (clustering), FIFO MLFNN	15 min	15 min	Flow	U	1	Y	N
SVR	Wu et al	2004	Current-time, historical average	3 min	3 min	Travel time	H	3	N	N
SVR	Vanajakshi and Rilett	2004; 2007	MLF NN (similar performance), historical average, current-time	2 min	1 hour at 2 minute intervals	Speed	H	3	N	N
Online-SVR	Castro-Neto et al	2009	GML (better in recurrent conditions), Holt exponential smoothing, MLP	5 min	5 min	Flow	H	7	N	N
Bayesian Network	Sun et al	2005	Random walk, Markov Chain	15 min (vehicles/hr)	15 min	Flow	U	4	Y	Y
Bayesian ADLM	Fei et al	2010	AR(2)	5 min	5 min	Travel time	H	1	N	Y
PCA-SVR	Jin et al, 2007	2007	ANN, ARIMA	15 min	15 min	Flow	U	58	Y	N
SVR-CACO	Hong et al.	2011	SARIMA	1 hour	1 hour	Flow (equivalent of passengers)	H	1	N	N
SSVRCIA	Hong et al.	2012	SARIMA, Seasonal Holt Winters, BPNN	1 hour	1 hour	Flow (equivalent of passengers)	H	3	N	N
Gauss SVR-CCPSO	Li et al.	2012	ARMA, BPNN, SVR, SVR-CPPSO	1 hour	1 hour	Flow (equivalent of passengers)	U	1	Y	N
Gaussian Processes Regression	Xie et al.	2010	v-SVM, ARIMA (v-SVM similar performance)	15 min	15min, 30 min	Flow	H	4	N	N

Key: Column headings: AA=Application Area (U=Urban, H=Highway); N=Number of locations modeled; S=Spatial data (are they included? Y=Yes, N=No); M=Missing data (are the accounted for? Y=Yes, N=No).

3.5.4.1 Variables

The traffic variable that is modelled largely depends on the available data. The majority of empirical studies focus on flow data (29 of 39 in table 3.3), usually collected from loop detectors. A smaller proportion of the literature (7 of 39) focusses on travel time forecasting. Aside from data availability, a reason that travel times are not as widely studied may be that observations are not instantaneously available and forecasts are lagged by the travel time value itself, necessitating longer range forecasts (van Lint, 2008; Liu, 2008)[10]. Furthermore, flows are directly observed whereas travel times and speeds often need to be derived (Whittaker et al., 1997) Nevertheless, travel times are the most understandable measure of road network performance and thus are very important, particularly in the context of traveler information systems.

In the majority of studies, past observations of a single variable are used to forecast its future values in an autoregressive framework, although some studies have focused on forecasting using a combination of variables (Liu et al., 2006b; Whittaker et al., 1997).

3.5.4.2 Aggregation and forecast horizon

Data aggregation typically varies between 5 and 15 minutes, although aggregations as low as 1 minute and as high as 1 hour are sometimes used. The majority of studies focus on one-step ahead forecasts only (27 of 39) with the step usually being equal to the forecast horizon. Abdulhai et al. (2002) argue that the forecast horizon should always equal the aggregation level, and that higher aggregations lead to smaller errors. However, it should be noted that when aggregating the data, one is creating a new series and forecasting a different quantity so the results at different aggregations are not directly comparable. Smaller errors are obtained on aggregate data because aggregation smooths noise in the data. This is the principal reason that data of very high temporal resolution are nearly always aggregated in the studies surveyed, i.e. from 30 seconds to 5 minutes. For those studies in which multi-step forecasts are carried out, horizons typically extend to around 1 hour. Beyond this point there is generally not enough local temporal information available to warrant the use of a sophisticated forecast technique. An exception to this is Ghosh et al. (2009) who forecast up to 12.5 hours ahead with good accuracy, which suggests a very stable recurrent traffic pattern in the dataset.

[10] This is discussed in more detail in chapter 4.

3.5.4.3 Application area and inclusion of spatial information

Although many of the models (23 of 39) incorporate spatial information, most are applied to highway data (26 of 39). The fundamental differences between urban and highway traffic have been described in section 3.4 and so are not repeated here, but it can be surmised that models that are appropriate for highway data are not necessarily appropriate for urban traffic. Most of the models that incorporate spatial information are applied to small datasets, typically consisting of fewer than 20 locations. There are few examples of applications to urban networks of the size that would be encountered in practice. Exceptions to this are Kamarianakis and Prastacos (2005) (25 links), Jin et al. (2002) (58 links) and (Min et al., 2009a; Min et al., 2010b) (50 links). Min and Wynter (2011) apply their MSTARMA model to 500 links but in the context of highways. The ways in which different models incorporate spatial information are discussed in more detail in section 3.6.

3.5.4.4 Treatment of missing data

Very few of the models (4 of 39) explicitly account for missing data in their specifications. The typical approach is to remove periods of missing data from the training data when building the model. This means that most of the models in the literature are ill equipped for dealing with missing data. Some exceptions exist, which are discussed in more detail in section 3.6.

3.5.4.5 Comparison with other models

Table 3.6 is a confusion matrix that compares the performance of different model types in the literature. For this review, the models have been separated into 7 categories, which are:

1 ARIMA type models: This category includes all models based on the ARIMA modeling framework, including seasonal versions and spatio-temporal extensions.
2 ANNs
3 Kernel methods
4 State-space
5 Bayesian networks
6 Nonparametric regression: This category includes k-NN approaches and standard kernel regression techniques.
7 Other: This category is reserved for models that do not fit into the other categories.

A nonzero entry $n_{i,j}$ indicates that model i outperforms model j in n separate studies. Nonzero entries on the diagonal of the matrix indicate that a model has been compared with a model of the same type. It can be seen that ARIMA type models tend to be compared with other ARIMA type models. Most examples of ARIMA type models outperforming other models are from earlier studies (Williams et al., 1998, Smith et al., 2002). A weak conclusion from examining the table is that nonlinear machine learning methods tend to outperform linear models. As the most widely studied machine learning method, ANNs appear have very good performance overall. However, kernel methods perform strongly compared with ANNs.

Table 3.4 – Confusion matrix of model performance

Model	ARIMA type	ANN	Kernel Methods	State-space	Bayesian Network	Nonparametric Regression	STM	Other
ARIMA type	5	2	-	1	-	1	-	1
ANN	3	4	1	2	-	-	-	
Kernel Methods	5	4	2	-	-	-	-	1
State-space	1	3	-	2	-	-	-	-
Bayesian Network	1	-	-		-	-	-	-
Nonparametric Regression	1	1	-	-	-	1	-	-
STM	1	-	-	-	-	-	-	-
Other		-	-	-	-	-	-	-

One of the difficulties in assessing the literature is the variety of settings in which models are applied, as evidenced by table 3.3. Rarely are models applied in an identical setting. One exception to this is the work carried out on the Athens road network (Stathopoulos and Karlaftis, 2003; Vlahogianni et al., 2005; 2007). Of the approaches used here, the genetically optimised modular network of Vlahogianni et al. (2007) yields the best results when compared to ARIMA and state-space methodologies, as well as other ANNs. Recent studies have shown that modified ANN structures yield further improvements in accuracy over the genetically optimised TDNN (Chan et al., 2013b). The work of van Lint and colleagues also suggests that ANNs are favourable over state space approaches, particularly on noisy data (Liu et al., 2006a). However, repeated mention is made in the literature of the excessively long training times in ANN models and their lack of interpretability compared with statistical models. Furthermore, van Lint (2006) makes a secondary comparison of 11 models of different types including ANNs, state-space models and SVR, and finds that SVR produces the lowest mean absolute relative error (MARE), although the application area is different in each study.

In a recent study, Chen et al. (2012) compare a range of forecasting methods in the context of traffic flow forecasting on highway data[11]. ARMA, ARIMA, SARIMA, SVR, FFNN, BN, k-NN and two naïve methods are tested. Each of the models is the basic implementation, so the study provides a good benchmark of performance. One step ahead forecasts are calculated at aggregation levels of 3, 5, 10 and 15 minutes. Contrary to the results reported in other studies, the ARMA model performs strongly, and marginally outperforms the nonlinear models at the 3 and 5 minute aggregations. The performance of the nonlinear models improves as the aggregation increases, but overall there is very little difference between the performance of ARMA, ARIMA, BN, SVR and ANN. Keeping the aggregation constant but forecasting at longer horizons may reveal greater differences between the models.

A characteristic of the data used in Chen et al. should be noted, which is rapid fluctuation from one time point to the next. This may be as much an artefact of the data collection system as a representation of the actual conditions on the road. This type of fluctuation is suited to ARIMA modelling, and may also explain why the SARIMA model performs poorly in comparison. In general, the similarity in performance of each of the models highlights the achievements of the traffic forecasting literature, in which significant improvements in accuracy are gained with bespoke model formulations. Another interesting finding from the study is that the performance of the models can be improved in most cases by removing the intra-day trend and forecasting the residuals.

3.5.5 Parametric versus machine learning methods in traffic forecasting – A summary

Karlaftis and Vlahogianni (2011) summarize the similarities and differences between statistical models and ANNs, which can be (in the most part) generalised to parametric and ML techniques as a whole. They cite five main differences; terminology, philosophy, goals, model development and knowledge acquisition, which make cross-pollination of research between the fields difficult. However, despite these differences there are also many similarities and often applying statistical or neural networks may result in the same model. For instance, a TDNN can be seen as having a nonlinear moving average structure (NMA). These parallels occur across the traffic forecasting literature; for instance, state space models are very similar to ARMAX models and reduce to ARIMA models in some cases (Stathopoulos and Karlaftis, 2003). It would be exceedingly difficult for a transportation

[11] This study is not entered in table 3.3 because it does not present new forecasting methods.

practitioner to choose between the available models even on pure performance in terms of model error, leaving aside other important factors such as parameter stability, explanatory power, causality and error distribution (Karlaftis and Vlahogianni, 2011).

Whatever the chosen model, the root of the problem lays in adequately capturing the underlying process that generated the data under study, whether it is explicitly through the parameters of a statistical model or implicitly through a kernel induced feature space or ANN. In spatio-temporal data this involves accounting for the spatio-temporal dependency between locations. If the processes behind the space-time distribution of the variable are understood, a state-space model is the most compelling approach (Heuvelink and Griffith, 2010). However, errors in the data collection process, insufficient spatial and temporal resolution and the presence of exogenous factors mean state-space models are not always appropriate, particularly in the urban setting.

Statistical models use theoretically well founded methods for measuring, incorporating and analysing spatio-temporal dependency structures and are very powerful if the model assumptions hold. However, in many complex real world datasets this is not the case, and space-time stationarity is difficult to achieve (Cheng et al., 2011). This leads statistical space-time models to tend towards mean values and perform less well in peaks (Vlahogianni et al., 2004). This is exacerbated in spatio-temporal data when attempting to fit a model to heterogeneous spatio-temporal structures, which might be part of the reason that statistical models are often not compared to other model types in the literature. The growth of the field of non-parametric methods has been based around the need to model complex, nonlinear datasets and they are well suited to the task. However, they can be susceptible to problems of overfitting and can be difficult to interpret. In the following section, the methods used in the literature for incorporating spatial information are discussed to gain further insight into the issue.

3.6 Neighbourhood selection in spatio-temporal models of road network data

When building a space-time model of network data, the researcher is faced with the task of determining how to incorporate spatial and temporal structure in the model. This can be referred to as spatio-temporal neighbourhood (STN) selection. The STN can be defined as the neighbourhood of spatial and temporal information that is used to forecast some quantity at a given location and time. The STN should only include spatial and temporal

information that is relevant to the point being forecast. To determine relevance, one may make assumptions about the physical process of traffic, or use a statistical measure of dependence such as a correlation measure.

The STN is incorporated into models in different ways depending on their type. In statistical models such as the STARIMA model, a spatial weight matrix is used to define the spatial arrangement of measurement locations (Pfeifer and Deutsch, 1980; Kamarianakis and Prastacos, 2005). This method is attractive because the weight matrix is model independent. Different models can be tested with the same weight matrix.

In ANNs, the STN may be defined in the internal structure of the model, and thus may not be interpretable. However, this drawback can be addressed by explicitly representing the spatial layout of the system in the hidden layer, which makes the internal function of the ANN more transparent (van Lint et al., 2005; van Lint, 2006). In Bayesian networks, the spatial structure of the data is represented by their conditional independence structure of the DAG. State space approaches use a physical model to represent the spatio-temporal evolution of traffic. In nonparametric regression methods and kernel methods, spatial information can be included as additional variables in the model. The spatial information to include can be controlled by a spatial weight matrix or distance matrix. Alternatively, feature selection methods can be used such as correlation analysis or Graphical Least Adaptive Shrinkage and Selection Operator (GLASSO) (Gao et al., 2011).

Independent of the ways in which different models incorporate spatial information, a review of the literature reveals that there are five categories of STN that vary in the level of detail they attempt to model, which are:

1) Globally fixed – The STN is fixed globally and does not vary spatially or temporally.
2) Temporally local – The temporal influence varies with time of day/traffic state, but the spatial influence is fixed.
3) Spatially local – The STN from location to location, but the temporal influence is fixed.
4) Spatially and temporally local – Both the spatial and temporal influences are local
5) Spatially and temporally dynamic – The spatial and temporal structure of the model evolves dynamically in time.

Within these categories, there are three distinct methodologies for STN selection:

1) Arbitrary – The STN is defined arbitrarily based on prior assumptions about dependency relationships.
2) Estimated apriori – The STN is estimated apriori through an exploratory data analysis step such as autocorrelation analysis.
3) Learned – The STN structure is learned from the data as part of the model training procedure.

In the following section, the literature is divided into these categories and the approaches are discussed in detail.

3.6.1 Fixed STN

The most straightforward way to incorporate the spatio-temporal neighbourhood in traffic forecasting models is to assume that its structure is fixed. This means that, for each location in the network, a forecast is made based on observations at a fixed number of previous time points and a fixed number of spatial neighbours. Within this, the neighbourhood can either be fixed arbitrarily, determined apriori, or learned directly from the data.

3.6.1.1 Arbitrary

Early work in traffic forecasting that uses spatio-temporal information incorporates it in an arbitrary manner. Okutani and Stephanedes (1984) employed Kalman filtering theory to predict short term traffic volume using spatial and temporal data. Due to computational limitations at the time, the study was limited to forecasting a single link based on three neighbouring locations. In the first spatio-temporal application of ANNs to traffic forecasting, Kirby et al. (1997) used the previous six data points of three upstream and one downstream location as inputs to their model.

In Whittaker et al. (1997), the topology of the highway is incorporated using a conditional independence graph. The current state of a location is necessarily defined by its previous state and the state of its upstream neighbours so the spatio-temporal neighbourhood is static. The strength of this approach is its foundation in traffic theory, but this is also a weakness as it is makes it inapplicable to signalised urban traffic data. More recently, Bayesian networks have been proposed that incorporate spatial information in a similar way to Whittaker et al. In a Bayesian network, a directed acyclic graph (DAG) is built to reflect the real life characteristics of the transportation network. Edges in the graph are directed forward in time and in a downstream direction. Usually, the structure of the DAG

necessitates that the measurement at a location depends only on its own previous values and those of its immediate upstream neighbours. Sun et al. (2004;(S. Sun et al., 2005) developed a Bayesian network for forecasting traffic flow. The spatial information is incorporated into the DAG based on the assumptions of the authors.

Similar approaches have been presented by Yu and Cho (2008) and Dong and Pentland (2009). In the former, both upstream and downstream locations are included in the DAG. An example DAG is given for a single link, but it is not clear how the DAG would be constructed for a large network. In each of these cases, the graphical structure is decided apriori based on perceived dependency relationships. Queen et al. (2007) also propose a Bayesian network for traffic flow forecasting on a motorway network in the UK, but use a different forecasting algorithm to the aforementioned studies. Only first order upstream flows are considered. In a further study, Queen and Albers (2008) show that the LMDM can be used to identify contemporaneous causal relationships between variables.

In the MST model of Ghosh et al. (2009), a multi-input multi-output model is presented that models multiple intersections on a transport network which are not necessarily situated on the same route or sharing the same path within the network. The cause and effect relationship need not be explicit which is advantageous when the precise relationship between locations is not known. Many real world urban sensor networks (including London's LCAP network) have low spatial resolution and a heterogeneous spatial structure that does mimic the physical structure of the road network. However, the contribution of neighbouring locations is modelled explicitly through the inclusion of flow observations at the single closest upstream location as an exogenous variable. This choice is arbitrary.

(M.-W. Li et al., 2013) apply Gauss-SVR trained with chaotic cloud particle swarm optimisation (CCPSO) to urban traffic flow forecasting. The model is tested on a single road section in Dalian City, China. Three upstream road sections are included in the model, which are chosen arbitrarily.

3.6.1.2 Estimated apriori

In many studies, the choice of spatio-temporal neighbourhood is estimated prior to model training through an exploratory data analysis (EDA) step. Dougherty and Cobbett (1997) used elasticity tests to select the most relevant data for forecasting, using a case study of flow, speed and occupancy data collected on the A12 motorway near Rotterdam, Netherlands. Kamarianakis and Prastacos (2005) built a STARIMA model to predict relative

velocity of traffic flow. The hierarchical ordering of road links is leveraged to build spatial weight matrices and this fixed spatial structure is used to provide inputs to the model. The fundamental assumption is made that downstream locations depend only on upstream locations and not vice versa, therefore the weight matrices are asymmetric. After differencing the data to achieve stationarity, the STN is determined through autocorrelation analysis.

In a related approach to Kamarianakis and Prastacos, Chandra and Al-Deek (2009) used cross-correlation analysis to determine the spatio-temporal neighbourhood, using data from 5 monitoring stations on the I-4 freeway in downtown Orlando. Contrary to the assumption of Kamarianakis and Prastacos, it is discovered that the influence of traffic comes from both directions.

Jin et al. (2007) use SVR with principal components analysis (SVR-PCA) to forecast urban traffic flow. PCA is used to extract temporal and spatial correlations among traffic flows observed at locations on the network. Forecasts are carried out on the transformed dataset of eigenflows. The traffic flows are then reconstructed from the forecast eigenflows. A significant benefit of the approach is that a single model is trained for the whole network, which is not usually the case with SVR models. However, the approach also has the drawback that the effect of the STN is not explicit.

More recently, Gao et al. (2011) used the graphical least absolute shrinkage and selection operator (GLASSO) for feature selection in the context of traffic flow forecasting. The GLASSO algorithm estimates a sparse graph from an inverse covariance matrix by applying an L1 (LASSO) penalty (Friedman et al., 2008). In their approach, Gao et al use GLASSO to define the STN of SCOOT (split cycle offset optimisation technique) flow detectors. The resulting STN is used to produce forecasts with an ANN model. GLASSO is similar in motivation to autocorrelation analysis, but has the advantage that it considers the influence of all the spatio-temporal information simultaneously. Tuning of the regularization parameter shrinks the elements of the inverse covariance matrix towards zero. Only those spatial and temporal lags with nonzero coefficients are input to the model. Furthermore, as it is separate from the forecasting algorithm, GLASSO could be used for feature selection for a range of models.

The approach shows promising results, but a number of points can be noted. Firstly, pairwise covariances between all locations and times (up to 6 time steps in the past) are considered.

This is reasonable with a small network, but unrealistic and unnecessary when the network is large. Incorporating some prior knowledge of the spatial structure in the data may be advantageous when working on the large network scale in order to avoid the necessity to calculate very large covariance matrices. Secondly, the features that are selected are static in time, which means the model does not take into account the fact that the dependency relationship between locations may change in time.

3.6.1.3 Learned

In many studies, STN selection is integrated in the model training procedure. This is often the case in ANN models because the internal structure of ANNs allows neighbourhood data to be incorporated as neurons. In the TDNN of Abdulhai et al. (2002) the temporal lag is determined dynamically by a genetic algorithm and the effect of including adjacent spatial locations in the model is considered. It is found that the model outperforms MLF ANN when the spatial neighbourhood is incorporated. The effect of the STN is more significant at shorter forecasting intervals, and the spatial effect diminishes as the interval increases towards 15 minutes.

Ishak and Alecsandru (2004) compare the performance of several types of neural networks under different conditions. They test the methods with different STN size and under different traffic conditions. They also compare models with short term and long term trend components and find that the long term trend becomes more important as the forecasting horizon increases. Additionally, it is suggested that the long term trend component is essential for modelling recurrent congestion while the short term trend can model short term stochastic variation (non-recurrent congestion). Prior knowledge of the process is used to constrain the size of the spatial neighbourhood to either 1 upstream, 1 downstream or upstream and downstream neighbours, and only the current and previous observations (t-1, noted as t-5 due to 5 minute data resolution) are considered. The optimal STN is then determined from this range in the training process.

In a more sophisticated approach, Van Lint et al. (2002) propose state-space neural networks (SSNN) for forecasting freeway travel times. The SSNN hidden neurons mimic the topological structure of the freeway, with one neuron for each road link. Although the maximum extent of the STN is determined by the extent of the case study area, the topology of the SSNN can be simplified by removing the hidden neurons (links) that do not contribute

to the solution, creating a parsimonious model. However, due to the architecture of ANNs, this structure is fixed during the training process.

Vlahogianni et al. (2007) recognise that the specification of the STN in ANN models for traffic forecasting has received insufficient treatment. To address this, a genetically optimized modular time delay neural network (TDNN) is proposed to predict traffic flow on a signalized urban arterial in Greece. The strength of the approach is the modularity of the system, whereby the inputs of the network are decomposed into two or more modules (road links in this case) that are independent of one another. Their outputs are combined using a gate unit that combines them into a single output in a predefined fashion. This allows each link to have its own look back period, delay and embedding dimension, which are determined by the GA. Although the authors claim the predictor is evolving, these variables are fixed once defined in the training process and the authors note that they may change if one were to train the model on different time periods. Therefore the model is not fully dynamic. An interesting finding is that the temporal embedding and lags determined by the system are different to those determined by autocorrelation analysis used in a previous study on the same network (Stathopoulos and Karlaftis, 2003). This raises questions about the ability of autocorrelation analysis to reveal clear and consistent dependency relationships.

Hu et al. (2008) propose a hybrid process neural network based on spatio-temporal similarities for short-term traffic flow prediction. Their modelling process comprises three steps. Firstly, a self-organizing map (SOM) is used to cluster freeway links based on the similarity of their time series. The second stage involves calculating the CCF between locations and choosing those neighbours with the peak value to be included in the forecast. Finally, a hybrid process neural network (HPNN) is trained which is composed of time varying process input signals, space aggregation, time aggregation, incentive threshold and incentive output. Wavelet transform is also used to extract the features of the time series data. Each spatial time series is fed into a unique module which learns its time varying process characteristics, then the outputs of these modules are fed into a non-time varying hidden layer which learns the forecast value of a site according to the contribution of its neighbours. The contribution of different sites to a forecast is allowed to vary. The model outperforms the same model trained with just upstream and downstream neighbours rather than neighbours determined by spatio-temporal mining. This finding is contrary to the rationale used for building many space-time traffic models that the closest spatial neighbours are those that provide most information and suggests the presence of a hierarchy of links.

Srinivasan et al. (2009) propose a neural network based approach to traffic flow forecasting that incorporates the principles of evolutionary strategy (ES). Upstream and downstream links are incorporated in the model, and downstream links are found to contribute more to the solution than upstream links. This is attributed to the temporal data resolution (15 minutes). It is not clear precisely how the spatial information is included, although it appears to have been selected as part of the model training procedure.

Zhu et al. (2009) present a space-time Kalman filtering approach for travel time prediction. A set of suitable adjacent links is selected using cluster analysis and used as inputs to the filter. The state transition matrix is then estimated by combining the spatial and temporal effects separately. This approach combines the approaches of learned and apriori STN determination.

Chan et al (Chan et al., 2013a; 2013b) integrate ANNs with sophisticated model selection techniques based on particle swarm optimisation (PSO). In both of these studies, the spatial neighbourhood is arbitrarily defined before the model training, but its contribution is optimised in the internal structure of the ANN.

3.6.2 Temporally Local STN

A temporally local STN is one in which the structure of the model varies with time, either by varying the composition of the spatial neighbourhood, varying the model parameters, or both. Most of the models that incorporate temporally local STN structures also incorporate spatially local structures, and they are discussed in sections 3.6.4 and 3.6.5. Here, those models that account for locality in time and not space are discussed. These models all take the approach of varying the model parameters with time, while the neighbourhood composition remains constant.

There are two approaches to time varying model parameters; either the model parameters vary smoothly with time, leading to one parameter combination per time instant; or they vary with traffic states, leading to one parameter combination per traffic state[12]. Of the former type, Fei et al. (2011) proposed a Bayesian dynamic linear model (DLM) for short term freeway travel time forecasting. The DLM sequentially revises the state of a priori knowledge of travel time based on newly available information. The model adapts to the changing characteristics of data by adjusting the system evolution noise level.

[12] Some of these models do not include spatial information, but are mentioned because of the way they treat temporal locality.

Also of the former type, Anacleto et al. (2013a) extended the LMDM of Queen et al. (2007) to accommodate the seasonal pattern in traffic flow data. This is accomplished using a time varying mean vector containing a mean flow level for each 15 minute period of the day. Anacleto et al. (2013b) extend this model to accommodate data from other sources using cubic splines. The method is able to better account for the heteroskedasticity in traffic data, and outperforms the same model with a fixed variance assumption.

Of the latter type, Hong et al. (2011) present an SVR model, trained with continuous ant colony optimisation, in which separate models were trained for the AM and PM peak periods. The effect of this division is not clear as it is not stated how well the algorithm performs if the data are not separated. In the same setting, Hong (2012) apply seasonal SVR with chaotic immune algorithm (CIA) for traffic flow forecasting. As well as dividing the data into AM and PM peaks, the data are seasonally adjusted before the SVR model is applied.

3.6.3 Spatially local STN

A spatially local STN is one in which the structure of the model varies with time, either by allowing the size of the spatial neighbourhood to vary in space, or by allowing the model parameters to vary across space, or both. Although all of the models in the literature that are applied to a single location could be considered spatially local, attention is paid here only to those models that explicitly account for spatial locality.

(Min et al., 2010b) propose a Generalised STARIMA (GSTARIMA) model to account for spatial heterogeneity in traffic data. In this model, spatially heterogeneous autocorrelation structures are captured by allowing the AR and MA parameters to vary by location. Although the method allows for spatially dynamic parameter estimates, the spatio-temporal structure of the model is fixed to an extent as the size of the spatial neighbourhood considered is the same for each location. Its temporal structure is also fixed and does not vary with traffic states or time of day.

Chan et al. (2012b) present an ANN model whose configuration is determined by the Taguchi method. In the Taguchi method, the temporal embedding dimension of neighbouring locations is selected as a design factor in the ANN training process. The embedding dimension is spatially local within the forecasting model for a single location. However, it is unclear how the model would scale to a large network based on the results for a single location.

3.6.4 Spatially and temporally local STN

A spatially and temporally local STN is one in which the model parameters and/or composition of the STN can vary in time and space. These models have been devised in response to the observation that global model structures cannot model the nonstationarity and heterogeneity present in traffic series.

3.6.4.1 Apriori

Relaxing the fixed structure of STARIMA and other statistical space-time models to deal with dynamic and heterogeneous data is an issue that has been tackled by a number of authors recently. One approach to this is to draw from a set of models that each pertain to a certain traffic state. This approach was taken by Min and Wynter (2011), who devised a multivariate (M)STAR model, based on the vector (V)ARMA model, whereby the weight matrix is selected from a pre-specified set of templates that reflect typical traffic states. Average speeds are used to calculate the number of links that can deliver their traffic to the current location within a given prediction horizon during these states, and these are used as the inputs to the model. Although the authors define just 6 templates in the empirical study, the methodology allows for any number of templates to be defined. Their approach shows impressive forecasting performance multiple steps ahead and the authors claim it is scalable to large networks.

An issue that is not discussed by Min and Wynter (2011) is the potential difficulty of identifying traffic states in real time. New traffic patterns must be classified into the correct template in order to be forecast successfully. This is likely to be more difficult in an urban setting, where traffic is affected by a diverse range of factors.

3.6.4.2 Learned

In the state-space model of Stathopoulos and Karlaftis (2003), cross-correlation analysis is first carried out to determine the candidate STN to train the model, and the best model is then learned from this parameter space. Cross-correlation is found to vary with time of day, so the day is separated into 6 periods and a model is trained for each. This results in a temporally and spatially local model specification. The approach requires strong and consistent correlations between detector locations, which may only be present in data of high spatio-temporal resolution. Furthermore, the method performs less well in unstable congested conditions when accurate forecasts are most important. The ANN method of

Vlahogianni et al. (2007)has since been shown to be more accurate on the same test data, despite having a fixed spatio-temporal structure.

Perhaps the most complete treatment of spatio-temporal dependency in a traffic forecasting model is the adaptive LASSO of Kamarianakis et al. (2012). The approach has a similar motivation to Min and Wynter, but the neighbourhood definition is learned rather than pre-specified. LASSO is a form of L1 regularised regression that is strongly related to GLASSO, but the feature selection and regression are carried out simultaneously. In the adaptive LASSO model, the traffic flow data for each link is divided into regimes using a threshold regression approach. Then, LASSO is used to automatically select the relevant data from the spatio-temporal neighbourhood and forecast the future traffic flows.

The approach shows excellent results, however, some drawbacks can be identified. Firstly, the approach eliminates the advantage of the related GLASSO approach of Gao et al. (2011) that any model can be used to forecast the traffic series once the relevant information in the spatio-temporal neighbourhood has been identified. The model is restricted to a linear regression model, and hence all nonlinearity must be accounted for in the regimes. Furthermore, as the authors note, it assumes abrupt transitions from one traffic state to the next. Finally, although the model is tested on an urban network, it is not tested on signalised road sections. The presence of signals would complicate the spatio-temporal neighbourhood identification procedure and, according to the authors, would result in increases in error of 10-20%. The performance on the urban highway network tested is poorer in terms of MAPE than that on two freeway networks.

3.6.5 Spatially and temporally dynamic STN

A spatially and temporally dynamic STN is one in which the spatial and temporal structure of the model changes dynamically according to the current traffic condition. This is perhaps the most ambitious strategy, and often requires a number of assumptions to be made in order to be successful. The models proposed thus far in this category all take similar approaches to vary the STN dynamically. The current traffic condition is used to determine the size and contribution of the STN to the current forecast value.

Min et al., (2009a) propose a dynamic form of the STARIMA model based on the idea of dynamic turn ratios. In their model, the traditional distance weighted spatial weight matrix used in STARIMA models is replaced with a temporally dynamic matrix that reflects the current turn ratios observed on each of the intersection of the road network. Thus the

weight matrix can be updated in real time based on current conditions. However, the method is highly specialised and the authors draw attention to some limitations: 1) it requires accurate intersection based flow data; 2) it performs less well on forecasting sites with highly variable flow, and; 3) it does not take into account the effect of traffic signals.

In a similar approach, Ding et al. (2010) propose a STARIMA model in which the elements of the spatial weight matrix are varied based on the current level of service (LOS). The LOS templates cover 6 ranges of average travel speed. Like Min and Wynter (2011), the spatial neighbourhood expands and shrinks according to traffic states, the difference being that Ding et al. propose to vary the states dynamically according to the prevailing traffic condition. It should be noted that the dynamic strategy could be taken in the approach of Min and Wynter, but is not considered in the study.

Cheng et al (2013) propose a localised (L)STARIMA model in which a dynamic spatial weight matrix is devised to account for spatial heterogeneity and temporal nonstationarity in the STN. The spatial weights are determined by the speed differential between the link in question and its upstream and downstream neighbours, and changes with each time step. The size of the STN is determined in the same way as Min and Wynter and Ding et al. The model is calibrated locally in space, and can be seen as a set of multivariate ARIMA models. The model outperforms STARIMA and a set of ARIMA models in travel time forecasting. However, the linear nature of the STARIMA model means that it cannot account for all the spatio-temporal autocorrelation in the data and the authors note that machine learning models may perform better within the same framework.

3.6.6 Summary of neighbourhood selection in traffic forecasting models

To date, a range of space-time traffic forecasting models have been devised that incorporate spatio-temporal information in a range of ways. The simpler models use a fixed STN. These models either assume that a stationary space-time process can be extracted from the data in a pre-processing step, or that the heterogeneity and nonlinearity in the data can be accounted for in the fixed (linear or nonlinear) model structure. The former assumption has been shown to be unrealistic; Cheng et al. (2013) compare their LSTARIMA model with both ARIMA and conventional STARIMA, and both the LSTARIMA and ARIMA significantly outperform STARIMA. The STARIMA model of Kamarianakis and Prastacos (2005) also fails to outperform a set of ARIMA models. This is because the global parametric structure cannot

model the nonstationary properties of the data, in particular the spatial heterogeneity. This is not surprising, and modelling spatially varying processes has been a key research area in the spatial sciences over the past 15 years, with models such as GWR growing in popularity (Fotheringham et al., 2002).

Nonlinear machine learning models that use a fixed STN implicitly assume that the nonstationarity and heterogeneity in traffic data can be modelled using a global nonlinear model structure. In ANNs, this requires sufficient variability to be captured in the internal structure of the model to describe all traffic states, without overfitting taking place. In pattern based models such as nonparametric regression and kernel methods, sufficient variability must be captured in the historical database of patterns. Furthermore, despite being nonlinear, both of these approaches assume fixity in the inputs to the model, and hence a fixed STN.

The recent trend in traffic forecasting models is toward structures that are local or dynamic in space and time. These models are better able to model the temporal nonstationary and heteroskedasticity, and the spatial heterogeneity of traffic data. The STN is usually incorporated using a time varying spatial weight matrix, with the exception of Kamarianakis et al. (2012), in which the spatial structure is defined in the model training process. To date, the application of spatially and temporally local or dynamic model structures in traffic forecasting has been confined to statistical modelling frameworks, and has not been assessed in the context of machine learning methods. A reason for this may be an assumption that machine learning models do not need local structures because they are nonlinear. However, this is not always the case. There has been a great deal of research, for example, into accommodating multiple scales into kernel methods such as SVMs (Cao, 2003; Pozdnoukhov and Kanevski, 2006). Multiple kernel learning (MKL) is a popular approach that allows kernels to be combined, enabling the fusion of data from heterogeneous sources (Sonnenburg et al., 2006). Introducing spatial and temporal locality into machine learning methods may realise significant performance gains.

However, caution must be exercised when using increasingly complicated model structures the risk of overfitting is increased. This is a particularly serious problem in machine learning methods. The spatially and temporally dynamic models that have been described thus far in the literature all make assumptions about the spatio-temporal evolution of traffic that are specific to the data under investigation. For example, Cheng et al. (2013) uses speed differentials to determine the weights of the spatial weight matrix. Some of the spatially and

temporally local models make similar assumptions. For example, Min and Wynter (2011) use predefined templates that describe the average relationship between spatial locations in different traffic states. Because of their basis in traffic theory, these approaches can be justified.

Other approaches, such as Kamarianakis et al. (2012) use feature selection to determine the STN. Therefore, the STN is determined entirely from statistical relationships in the data, with no assumptions made about the physical process of traffic. Such approaches rely on the true relationships being discovered, and remaining consistent throughout time. If relationships are present in the training data but do not persist in the future then this type of method may encounter problems. There is a clear trade-off between the detail that can be modelled and the generalisation ability of the model to future data. The superior performance of spatially and temporally local statistical models over globally fixed statistical models suggests that accounting for locality is necessary, but the extent to which this is possible depends on the strength of relationships in the data. In any case, it is sensible to incorporate theory to guard against identifying statistical relationships that are theoretically implausible.

3.7 Missing data treatment in traffic series

In this section, treatment of missing data in traffic series is discussed. Missing data are a serious problem in sensor networks as they result in incomplete space-time series. This complicates the application of many spatio-temporal forecasting models, which has been recognised as a problem in various application areas including traffic forecasting, environmental monitoring (Glasbey, 1995; Smith et al., 1996; Smith et al. 2003), video image reconstruction (Kokaram and Godsill, 1996, 1997, 2002; Kokaram, 2004; Kokaram et al., 1995a, 1995b) and hydrology ((Amisigo and Van De Giesen, 2005)). Missing data make accurate forecasting difficult as gaps in data collection necessitate the use of longer forecast horizons in the absence of other information. They can have particularly severe consequences for statistical models because many of them, including ARIMA type models, require a complete series to ensure the proper ordering of the temporal dimension and calibration of parameters. If there are large amounts of missing data in the training set then there may not be enough data available to train the model. Furthermore, if data are MNAR then important information about certain traffic states may not be accounted for. This latter problem affects the application of all forecasting models. For example, in pattern based models there may not be enough patterns in the training set to describe a certain traffic state.

3.7.1 Types of missing data

The level of missing data in a dataset, or its missingness, is defined as the proportion of records for which there are observations. Missing data does not refer to locations and times that are not covered by the sensor network, which is an issue of sampling design. Rather, it refers to those locations and times where observations should have been made but were not recorded. Missing data has different implications depending on its type. There are three types of missing data; missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR), which are described in turn in the following subsections.

3.7.1.1 Missing completely at random

When data are MCAR, the probability that an observation x_i is missing is unrelated to the value of x_i or to the value of any other variables (Howell, 2007). The impact of MCAR data is usually not severe. For example, in the context of space-time series, MCAR data are randomly dispersed in the series so the probability of large gaps in data collection is low, provided that the overall missing rate is low. Therefore there is usually enough local spatio-temporal information available to avoid significant decreases in forecasting performance. The implication may be, for example, that a forecast may not be available for a single period, or a multi-step forecast may be required.

3.7.1.2 Missing at random

The term MAR is used to describe data that are not MCAR, but whose value is unrelated to the value of x_i after controlling for other variables. In other words, it can be completely described by variables observed in the dataset (Zacarias and Andersson, 2011).

3.7.1.3 Missing not at random

Data that are MNAR are those for which the probability that an observation x_i is missing is systematically related to the hypothetical value of x_i that is missing. In the context of traffic data, Qu et al. (2009) describe MNAR data as those where the missing data have some pattern, possibly caused by long term sensor malfunction. This definition is not completely clear. The incidence of long term sensor failures across the road network may or may not be random. For instance, if a particular set of sensors often fails, then the probability of the data being missing is related to the sensor collecting the data and the data is MNAR.

However, if the probability that a sensor will experience long term failure is not related to the sensor itself, then this missing data can be seen as MAR.

However, if a sensor functions poorly under certain conditions, such as during congestion, low light conditions or during precipitation, then the process being observed under these conditions will be underrepresented in the data and the data are MNAR. MNAR data have the most serious implications from a modelling perspective as they induce bias in models, and can lead to a lack of available data for forecasting under certain conditions.

3.7.2 Methods for dealing with missing data in traffic series

The presence of missing data can be dealt with in one of two settings. Either it can be dealt with offline, which is known as imputation, or it can be dealt with in real time as part of a forecasting framework. Various offline imputation methods have been used in the spatio-temporal modelling literature, many of them based on principal components analysis (Smith et al., 1996) or expectation maximisation (Schneider, 2001; Smith et al., 2003). However, there has not been a great deal of research that has focussed specifically on dealing with missing data in a real time setting, where its presence necessitates long range forecasting. One application domain where researchers have begun to tackle this issue is transport (Whitlock and Queen, 2000; van Lint et al., 2005). This has been motivated by the frequently high levels of missing data that are present on traffic monitoring networks, and the need for accurate traffic information for operational purposes. The literature on missing data in traffic series can be separated into four categories:

1) Univariate Imputation – Data are imputed based on the traffic series of the location of interest
2) Multiple Imputation - As 1), but multiple imputation methods are used to improve consistency of results.
3) Spatio-temporal Imputation – Spatial and temporal information are used to impute missing values
4) Integrated forecasting – the treatment of missing data is explicitly dealt with in the forecasting model.

These four categories are discussed in turn in the following subsections.

3.7.2.1 Univariate Imputation

The imputation methods employed in practice have traditionally been simple univariate time series or factor based approaches that do not account for the dynamics of the real traffic situations (Zhong et al., 2004a). In recent years, a number of more sophisticated approaches have been developed that attempt to improve estimates. The studies of Zhong et al. (2004a; 2004b) and Sharma et al. (2004) compare the performance of factor approaches and ARIMA with genetically trained time delay neural networks (TDNN) and locally weighted regression. The models are applied to hourly flows from permanent traffic counters (PTCs) on the highway network in Alberta, Canada. The best performing model is a seasonal local regression model trained by a GA that makes use of data from either side of the failure.

A subsequent study by Zhong et al. (2006) proposes a simple imputation method based on pattern matching using data from before and after the failure. The method extracts a candidate set of normalised traffic patterns from the historical dataset and selects the best fitting pattern based on a minimum squared error (MSE) between the candidate pattern and the pattern under study. This best fitting pattern is then used to impute the data. The method is tested on traffic count data collected using permanent traffic counters (PTCs) on the ATR C002181 in Alberta, Canada and is found to outperform factor approaches, ARIMA and exponential moving average. The use of a single traffic pattern for imputation may be appropriate for the smooth, hourly data used in the study, but is unlikely to be sufficient when applied to data of higher temporal resolution that is corrupted by noise. This has been recognised by (Liu et al., 2008) who proposed a k-NN method for imputation of missing traffic data during holiday periods. To determine the k neighbours, a state vector is defined, augmented with historical averages (after Smith et al., 2002). To take into account the ranking of the neighbour set, the weights vary inversely with distance from the state vector. The results of Zhong et al (2004b; 2006) highlight the effectiveness of simple pattern based approaches for imputation.

(Qu et al., 2009) propose probabilistic principal components analysis (PPCA) for urban traffic flow imputation. The authors note two desirable properties of PPCA. Firstly, it uses only the major information and avoids overfitting faced by spline methods and regression methods. Secondly, the model parameters can be computed directly from the data via efficient eigenvector decomposition. The method can also be used to identify outliers or unusual patterns in the data. Components that cause a notable increase in variance and covariance in the original variables represent extreme values. The PPCA method is tested on 5 minute

flow data from loop detectors in Beijing, China and is found to outperform spline and historical imputation methods, particularly when the ratio of missing data is high. It also produces imputed values that are statistically consistent with the distribution of flow data.

3.7.2.2 Multiple Imputation

The methods described thus far all make a single imputation for each missing point. Their effectiveness is assessed by comparison with validation data. However, in real situations where the data are actually missing, the validation data are not available. Therefore, having some indication of the reliability of the estimated data can be important. In light of this, multiple imputation is a common approach. Ni et al. (2005) present a multiple imputation method for imputing traffic flow. The idea behind multiple imputation is to simulate multiple random draws from a population in order to estimate an unknown parameter. The expectation maximisation (EM) algorithm is used to generate maximum likelihood estimates of the missing values and these are used as inputs to a data augmentation (DA) procedure which is run k times. The whole process is repeated n times to produce n sets of imputed data. The consistency of the missing data estimates can then be examined by analysing the distribution of the n sets.

Wang et al. (2008) propose an alternative approach that combines non-parametric regression techniques with multiple imputation. k-NN is used as the imputation algorithm, with the historical dataset first being categorised into free-flowing, moderately congested and highly congested patterns. The current traffic pattern is then assigned to one of these categories prior to the k NNs being computed. A second method further classifies the data into different sub-segments based on road characteristics. The procedures are carried out multiple times to impute travel times on 10 roadside detectors on the I-70 in Maryland, USA and are found to outperform mean substitution and Bayesian forecasting methods. In this method, spatial information is included through the grouping of sensors into sub-segments; however, there is no discrimination between patterns collected on the sensor in question and the adjacent sensors.

3.7.2.3 Spatio-temporal approaches

Univariate techniques can be very effective but their applicability decreases with the length of the period of missing data. Many imputation methods assume that missing data are sparsely located within the series so that neighbouring temporal information can be used for

univariate forecasts. This is an unrealistic assumption as sensors often malfunction and stop collecting data for several consecutive time periods. Surprisingly, the use of spatial information in imputation and forecasting of traffic under missing data has not received much attention in the literature, although there have been recent attempts to address this. L. Li et al. (2013) used kernel probabilistic principal components analysis (KPPCA) for spatio-temporal traffic flow imputation. The model assumes that traffic flows at the same time collected on previous days are implicitly correlated. Furthermore, correlations are implied between these values at different detectors. To account for the spatio-temporal dynamics of traffic flow, flow values from the upstream and downstream detectors are included at times t+1 and t-1 respectively. The model is tested on highway data from the PeMs dataset, and outperforms standard PPCA, but it is noted that PPCA is significantly faster. The rigid spatio-temporal neighbourhood choice is justified in this case because the spatio-temporal resolution of the data is very high and the flows are uninterrupted, but this is likely to need further consideration in an urban setting.

Zhang and Liu (2009a; 2009b) applied LS-SVM to the imputation of traffic flows in urban streets. The Pearson coefficient is used to determine correlated traffic flows at other detectors, which are used to form a state vector as input to the model. The performance is assessed on loop detectors at 3 neighbouring intersections with missing data rates of 20% to 50%, with 10% increments. Multiple imputation (MI) based on expectation maximisation/data augmentation is used as the comparison method. LS-SVM is found to outperform MI at all missing rates. This is an important finding as it demonstrates that point estimates can yield higher accuracy than MI methods.

3.7.2.4 Forecasting approaches under missing data

The imputation methods discussed thus far are not designed to operate in real time. Therefore, they are not applicable in time critical operations that rely on timely data, such as traffic control systems and ATIS. In such systems it is necessary to be able to deal with missing data effectively in order to maintain performance. It was stated in section 3.5.4.4 that most forecasting models do not account for the presence of missing data, and generally make the assumption of complete series. If data are missing in the historical dataset then the affected data are usually discarded. This is a necessary step for model validation; however, it leads to the development of forecasting models that are only applicable in situations where data are available. This drawback has been recognised in the literature

recently, and a number of methods have been developed that account for missing data in real time.

In their comparative analysis of forecasting approaches, Chen et al. (2003) investigate the effect of PPCA for temporal imputation of missing traffic flows in a real time setting. Assuming a forecast is needed at time $t + 1$ and the data point z_t at time t is missing, z_t is first imputed using PPCA before being fed into the forecasting algorithm. Data are assumed to be a mixture of MCAR and MAR, based on the definition of Qu et al. (2009). Data that are MNAR are not considered. It is claimed that PPCA mitigates the problem of missing data when the missing ratio is less than 25%. However, a comparison is not provided with multi-step forecasts or simpler imputation schemes.

The approach of Chen et al. (2003) is requires a separate algorithm to be used for forecasting. There are a small number of examples in the literature of spatio-temporal forecasting models that explicitly take into account missing data. Whitlock and Queen (2000) propose a dynamic graphical model for forecasting highway traffic flows under missing data. In their approach, the traffic flow at a measurement site is considered to be independent of all other sites given its parents, which are its upstream neighbours. Methods are developed for dealing with situations where the data from one or more parents and/or children are missing. It is found that strong apriori knowledge of the missing series, which can be assumed in most cases, results in good performance. However, this is dependent on the regression parameters being fairly constant over time, which may not be the case in an urban setting.

Van Lint et al. (2005) propose a state space neural network (SSNN) that is robust to missing data. It is discovered that training the SSNN with perfect data reduces its capacity to deal with missing data. By incorporating simple imputation schemes such as spatial interpolation and exponential smoothing, the algorithm becomes less sensitive to missing data, despite slight changes in the statistical properties of the input data. The method is shown to outperform SSNN with no data imputation on simulated (FOSIM) data with missing rates of up to 40%. The method is also tested on real highway data taken from the MONICA system in the Netherlands and is shown to outperform the online estimator being used at the time (albeit a simple method). This method highlights the importance of effective imputation, and it is likely that its accuracy would be improved further if more accurate techniques were initially used to impute the missing data in the training set.

In assessing the performance of their IPSO ANN model, Chan et al. (2013b) test four missing data scenarios, based on two contingencies: 1) The sensors at a particular location have failed and no data are collected, and; 2) The sensors at a different location are functioning incorrectly and produce random white noise with a value between zero and 100. Based on these contingencies, the scenarios are: 1) complete data (neither 1 nor 2); 2) contingency 1; 3) contingency 2, and; 3) contingencies 1 and 2. The model is tested extensively against other neural models, including the models of Vlahogianni et al. (2005) and is found to exhibit superior performance. The improvement in performance under missing data is achieved due to the time varying structure of the algorithm.

3.7.3 Summary of missing data treatment in traffic forecasting

A range of imputation models have been described in the preceding subsections with varying degrees of sophistication, ranging from simple univariate averaging methods to complicated spatio-temporal neural network architectures which are capable of operating in a real time forecasting setting. It can be concluded that univariate methods usually perform well when the length of missing data is short but that spatio-temporal models are more effective when extended periods of missing data are present. MI methods have benefits in terms of interpreting uncertainty. However, in a real time forecasting setting, multiple imputation methods are not appropriate as they are computationally intensive. Furthermore, Zhang and Liu (2009a; 2009b) demonstrate that a single imputation approach based on LS-SVM can produce more accurate estimates than MI. What are required are forecasting methods that explicitly account for missing data in the model design.

The studies mentioned in section 3.7.2.3 represent progress towards achieving this goal. However, they have all been applied to highway data, where traffic flows are uninterrupted. In this setting, the spatio-temporal relationship between measurements at neighbouring locations tends to be strong. In an urban setting, it is more difficult to elicit this kind of relationship from data because of the factors discussed in section 3.4. This is exacerbated when the sensor network is spatially sparse, in which case the failure of a sensor can cause large spatial and temporal gaps in data collection. Methods such as Zhang and Liu (2009a; 2009b) might not perform as strongly in this setting as the correlation between traffic observations at adjacent locations would not be as strong. How to deal with this situation in the context of forecasting has not been fully addressed in the literature, and forms part of the focus of this thesis.

3.8 Chapter summary: Achievements and challenges in spatio-temporal traffic forecasting

In this chapter, the state of the art in spatio-temporal forecasting of traffic data has been reviewed. The chapter focusses on three key areas: 1) The range of methods that have been used for traffic forecasting to date; 2) Methods for accommodating the STN in spatio-temporal models of traffic data, and; 3) Methods for dealing with missing data in traffic forecasting models. Based on the review, there are some key achievements and challenges that can be noted. The achievements pay tribute to the outstanding work that has taken place in the traffic forecasting literature that has led to its current state of development. The challenges outline some tasks that still remain to be tackled, and motivate the work carried out in this thesis.

3.8.1 Achievements:

- **Forecasting of highway data:** The literature on forecasting of highway data is well developed, and a range of sophisticated modelling techniques have been applied to various traffic variables including flows and travel times. Because of the uninterrupted nature of highway traffic, traffic flow theory can be used to incorporate spatial information either directly or indirectly. The methods applied to highway data have also been implemented in practice, benefitting transport authorities. Methods exists for forecasting highway networks of hundreds of links (Min and Wynter, 2011).
- **Application of ANN models:** A great deal of research has been conducted into ANN models in the traffic forecasting literature, which has led to some notable achievements. Firstly, representing the spatial layout of the road network as neurons in the hidden layer has greatly improved the interpretability of ANN models, which have traditionally been viewed as black box. Secondly, improved training methods have helped to reduce the problems of local minima and poor generalisation ability that ANN models typically face.
- **Modelling local/dynamic structures in spatio-temporal network data:** Because large spatio-temporal traffic datasets have been available for some time, there are many methodological advances in the traffic forecasting literature that can be applied to other types of network. For example, the local and dynamic statistical models of Min et al. (2010b, 2009b), Min and Wynter (2011), Kamarianakis et al. (2012) and Cheng et al. (2013) have the potential to be applied widely to other

networks. The fields of network science, spatial analysis and econometrics, amongst others, could benefit from these achievements.

3.8.1.1 Challenges:

- **Modelling urban networks:** Much of the work carried out thus far focuses on flow data with high spatial and temporal resolution, often collected on freeways. There has comparatively less research into spatio-temporal models for large scale modelling of urban transportation networks. Furthermore, there has been little research into modelling networks where the sensor network is spatially sparse and the dependency relationships are unclear.
- **Dealing with missing data:** Although missing data has been acknowledged as a major issue in transportation research, there has been relatively little research into how to deal with it in the context of forecasting, where large spatial and temporal gaps may appear in the series. The approaches developed thus far have all been applied to highway data.
- **Incorporating locality and dynamics into machine learning models:** Much of the recent progress in traffic forecasting has stemmed from incorporating locality and dynamics in linear statistical models. However, there is strong evidence in the literature that nonlinear approaches such as ANNs and kernel methods are better at modelling the nonstationary properties of traffic data. To date, there has been no research into incorporating local structures into machine learning models.

Chapter 4 – The London Congestion Analysis Project

4.1 Overview of Chapter

In this chapter, the dataset used in this thesis is described. Firstly, in section 4.2, the London Congestion Analysis Project (LCAP) is introduced, which is a travel time dataset collected on London's road network. Following this, in section 4.3 a data quality analysis is carried out. In section 4.4, the data cleaning and preprocessing steps are described, which result in the dataset used in the remaining Chapters of this thesis. This dataset is summarised in section 4.5.

4.2 The London Congestion Analysis Project

The dataset used in this thesis is the LCAP, which is a link based travel time dataset collected using a system of automatic number plate recognition (ANPR) cameras installed on London's road network. The ANPR system is maintained by Transport for London (TfL), its main use is vehicle monitoring for operational purposes such as enforcing the Congestion Charging Zone (CCZ) and policing restricted areas such as bus lanes, red routes and hatched areas. Collecting travel time information is a secondary use of the system, but is of vital importance to TfL's Road Network Performance and Research (RNPR) team, who use it to measure the performance of key routes on the network. At the time of writing, the LCAP is the most extensive spatio-temporal traffic dataset available to TfL in usable form[13].

4.2.1 Travel time data

TTs are a very important measure of road network performance as they have an intuitive interpretation. Anyone can understand the significance of a 30 minute increase in their commuting time, but understanding the effect of an increase in flow of 100 or 1000 vehicles per hour on a particular road section is more difficult. Reliable forecasting of travel times is of great practical importance as forecasts of future travel times allow road users to reduce

[13] Although a significant proportion of London's traffic signals have Split Cycle Offset Optimisation Technique (SCOOT) detectors installed, the antiquated system does not allow easy extraction of traffic variables. Work is currently ongoing to validate SCOOT data against automatic traffic counter (ATC) data.

the margins of uncertainty involved in planning their activities (Ettema and Timmermans, 2007). Ettema and Timmermans (2006) demonstrate that overestimation of the variability in travel times leads to unnecessary delay, and that the provision of day-specific travel time information leads to a significant reduction in scheduling costs. In more uncertain environments such as urban road networks, the value of travel time information is even greater as uncertainty cannot be reduced by experience (Z. Sun et al., 2005). Furthermore, TT forecasts can enable road users to modify their route to avoid certain road sections that are likely to be congested during their journey.

The TT of a vehicle can be defined as the time taken to travel between two fixed points, for example, between one intersection and the next, or between home and work. Formally, the *TT* of an individual vehicle can be expressed as:

$$TT = t_2 - t_1 \tag{4.1}$$

Where t_1 is the time at which the vehicle began its journey and t_2 is the time at which it completed its journey. For the purposes of this thesis, vehicles are considered to traverse *links*. A link is a pair of points $l = \{l_1, l_2\}$, $l \in L$ where l_1 is the start point of the link, l_2 is the end point of the link, and L is the set of links for which travel times are available. In this case, l_1 and l_2 are successive ANPR camera locations on a route and a vehicle may traverse several links on a journey. The link travel time ($TT_{l,i}$) of vehicle i on link l is calculated as the time taken to travel from l_1 to l_2 (see figure 4.1). Individual TT observations can be aggregated within time intervals. The aggregated link travel time $TT_l(p)$ for time period p is the arithmetic mean of all vehicles that traverse link l during time period $p = [t, t + \tau]$, where τ is the time interval, and is calculated as (Liu, 2008):

$$TT_l(p) = 1/N \sum_{i=1}^{N} TT_{l,i}(t^*), t^* \in p \tag{4.2}$$

Where N is the total number of vehicles either arriving at l_2, or departing from l_1, during p, t^* denotes the time at which vehicle i departed/arrived. The former are known as arrival TTs and the latter as departure TTs. Departure TTs are preferred over arrival TTs in traffic engineering as they give a measure of the TT for a driver entering a road link at a given time instant (Liu, 2008). The LCAP data are 5 minute aggregated *departure TTs,* providing 288

observations per link per day. No information is available regarding the distribution of individual link travel time measurements that comprise each observation. However, the frequency (number of vehicles N) of each observation is available.

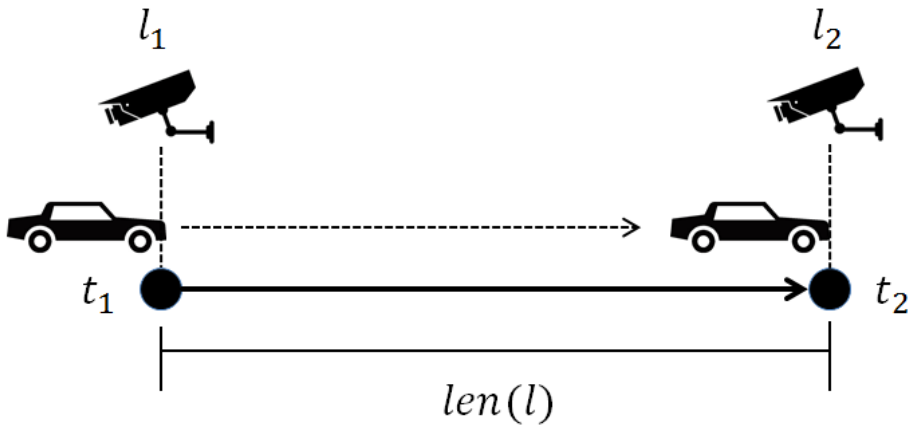


Figure 4.1 – Observing travel times using ANPR: A vehicle passes camera l_1 at time t_1, and its number plate is read. It then traverses link l and passes camera l_2 at time t_2 and its number plate is read again. The two number plates are matched using inbuilt software, and the TT is calculated as $t_2 - t_1$. Raw TTs are converted to UTTs by dividing by $len(l)$, which is the length of link l.

The drawback of departure TTs in a real time setting is that observations do not become available at the time instant they refer to, but once the journeys that began at that time are completed. Therefore, there is a time delay between the forecast time instant and the arrival of data for that time instant, which is approximately equal to the TT on that link. For example, if the individual TT of a vehicle entering a link at 9 AM is recorded as 5 minutes, then the observation will arrive at 9:05 AM. The implication of this in forecasting is that the latest observed travel time must be subtracted from the forecast horizon to give the true forecast horizon.[14] Clearly, the longer the link, the higher the individual TTs will be and the further ahead forecasts must be made in order to be useful. This problem is *exacerbated during congestion*, when observations may not become available for extended periods of time. Furthermore, when using a spatio-temporal approach, different information pertaining to the same time period will become available at different times due to variable link lengths and traffic levels. Multi-step short term forecasting (i.e. 30 minutes to 1 hour ahead) is thus

[14] An aggregate TT measurement can be recorded for departure time p once the first vehicle that departed at departure time $p + 1$ has been recorded.

very important when using TTs compared with flows or densities where point observations are usually made.

4.2.2 Unit Travel Times

Raw TT observations are scale dependent as they depend on the length of the link. The link lengths on the LCAP network vary significantly, making travel times on adjacent links incomparable. Therefore, the TTs are transformed into unit (U)TTs by dividing by the link length:

$$UTT_l = TT_l / len(l) \tag{4.3}$$

Where $len(l)$ is the length of link l (see figure 4.1). On the LCAP network, the TTs are measured in seconds and the link lengths in metres, so the unit of the UTTs is seconds per metre (s/m). UTT is the reciprocal of speed, which is an alternative measure of link performance. UTT is preferred over speed in this thesis because it provides an intuitive measure of delay; the higher the UTT the lower the link performance. The effect of the transforming UTTs to TTs is shown in Figure 4.2. In figure 4.2 a), the time series of ten randomly chosen LCAP links are shown over the course of a week. The y-axis shows the journey time in seconds for each of these links. It can be seen that the journey times vary significantly between links. Figure 4.2 b) shows the same set of links, but the raw TT observations (seconds) are transformed to UTTs. Comparing the two figures, the effect of the transformation is evident. Casting a cursory glance over the raw data, links 1757 and 1825 would appear to be more congested than the other links. Converting their values to UTT demonstrates that they actually behave similarly to some of the other links. Link 1829 is revealed as the most congested of the links over the course of the week.

This transformation is a form of normalisation, which is important in the context of spatio-temporal analysis as it brings the data of all the links into the same range, allowing their performance to be compared. UTT is a good measure to use in the context of travel time modelling because it has a physical meaning within traffic engineering. However, it would be possible to normalise the data in other ways. For instance, as well as calculating speeds, one could standardize TTs to produce z-scores, or normalise them to fall within a set interval, for example [-1,1]. Henceforth, *all analysis is carried out using UTT unless otherwise stated.*

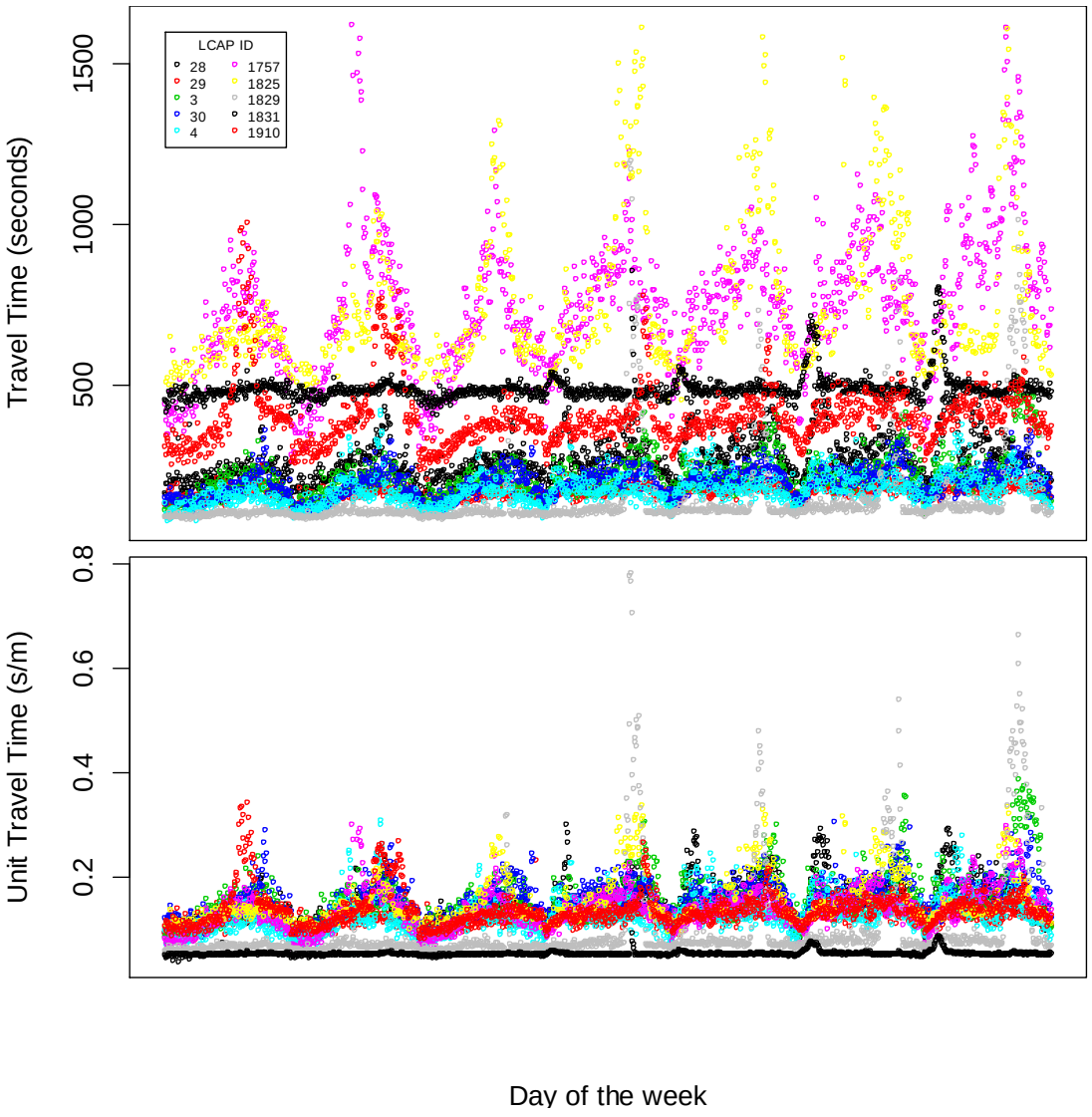


Figure 4.2 – Example of journey time data collected on the LCAP network, a) raw journey times b) unit journey times.

4.2.3 Spatial extent of the LCAP

The location of the LCAP network in London is shown in figure 4.3 a). In total, there are 758 links for which data are available. In outer London, the LCAP coverage is sparse, but in the central area within the congestion charging (CC) zone it is more dense (figure 4.3 b)), and roughly coincides with the network of A roads. Based on theories of the power law distribution of traffic flow on road networks, it is expected that the coverage of the LCAP network in the central area captures a significant proportion of the journeys taking place in the city centre (Jiang, 2009). This is not the case in outer London, where it only accounts for certain routes.

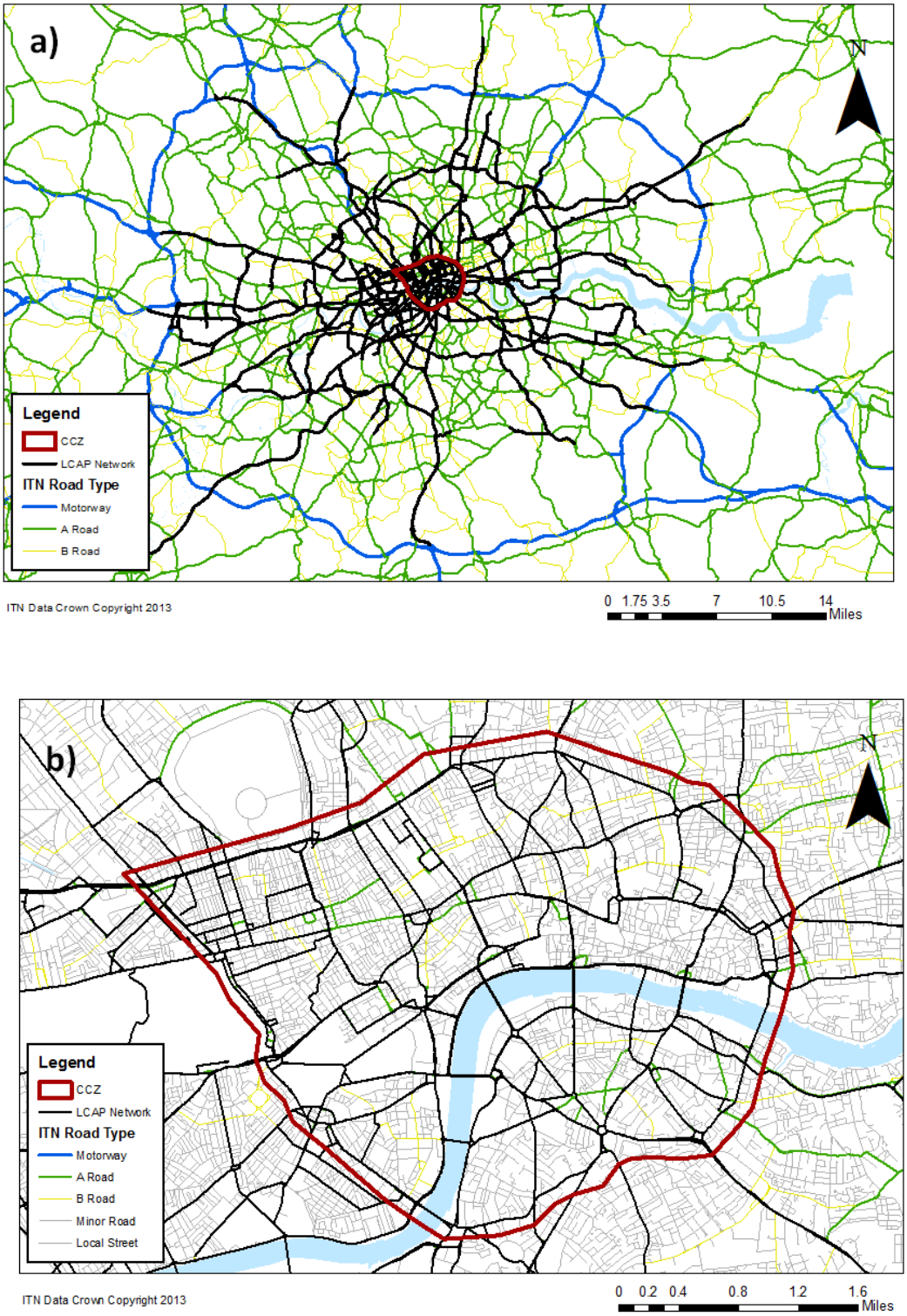


Figure 4.3 – The ITN and LCAP Networks: a) In the Greater London area; b) Within the CCZ

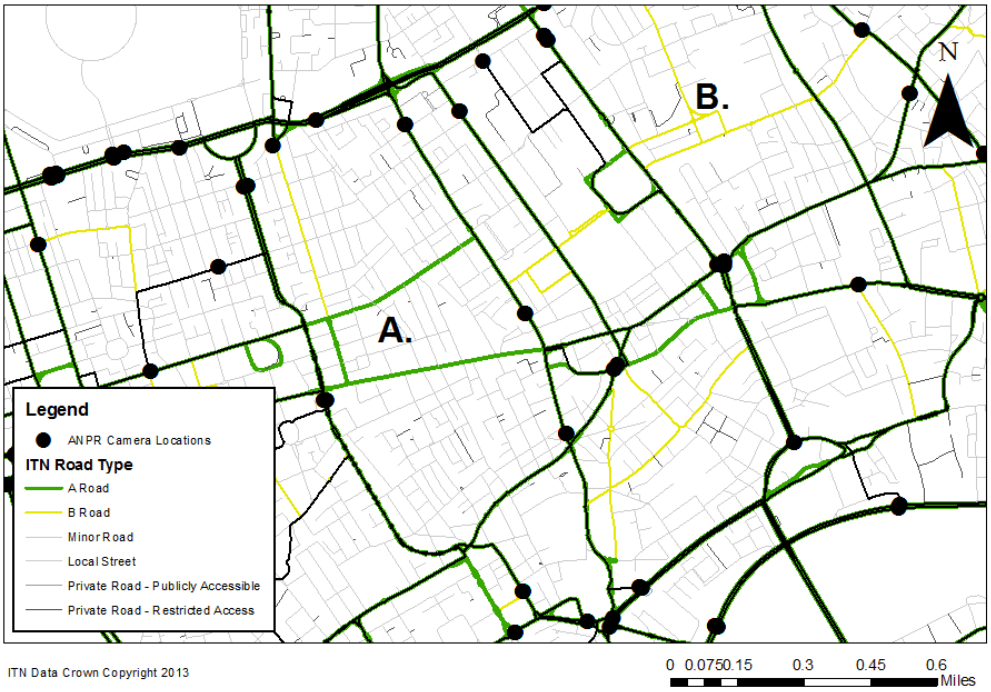


Figure 4.4 - Closer view of spatial coverage of LCAP within CCZ

Figure 4.4 shows a closer view of the coverage of the LCAP within the CCZ. Although covering most of the A roads, the coverage is not complete. For example, in figure 4.4 A), a set of LCAP links run in parallel connected by two A roads that are not part of the sensor network. Similarly, at figure 4.4 B) there is a set of parallel LCAP links connected by B roads that are not part of the sensor network. Beyond this, there is a dense network of minor roads that forms connections between LCAP links. Similar situations are seen across the sensor network. The general picture is one of a spatially sparse sensor network existing on a dense physical road network. There is no reason for journeys to take place only on the LCAP network, and flows can and will occur between LCAP edges along other routes.

4.2.3.1 Link Lengths

The length of individual links affects the nature of the travel times that are observed on them. Longer links smooth the stochastic variation caused by the presence of signals and the heterogeneity in individual driver behaviour. Shorter links capture more of this variation, but also capture more local traffic patterns and are more likely to be correlated with their spatial neighbours, provided they are also short. Furthermore, the delay in the observation of departure travel times is longer for longer links. The lengths of the LCAP links depend on the positions of the ANPR cameras, and vary widely. The maximum, minimum and mean link

lengths are 22576, 105 and 2735 metres respectively. Figure 4.5 shows a histogram of the link lengths. The link length distribution appears to be approximately lognormal.

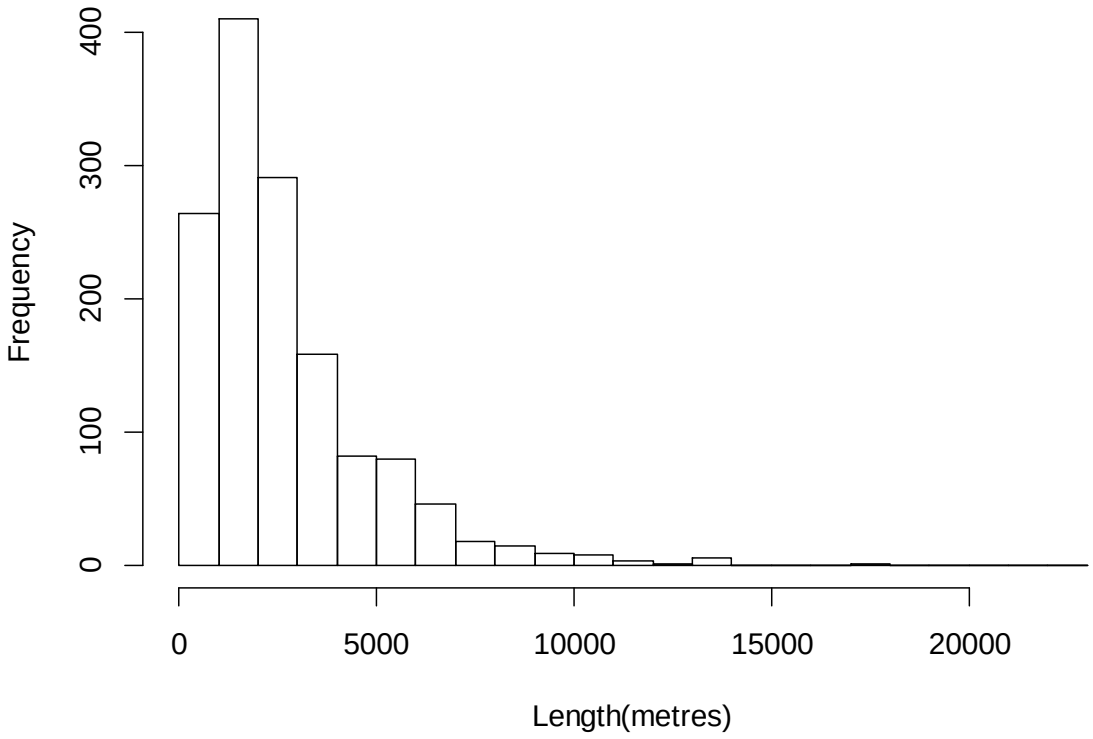


Figure 4.5– Histogram of link lengths

Figure 4.6 is a choropleth map of the link lengths on the LCAP network. The darker blue colours represent longer links, while the lighter blue colours represent shorter links. There is a general spatial pattern of longer links on the periphery of the study area, covering longer stretches of road. Links are generally shorter in the central area, where the density of the network is also higher. This reflects both the physical properties of the road network and the strategic priorities of TfL.

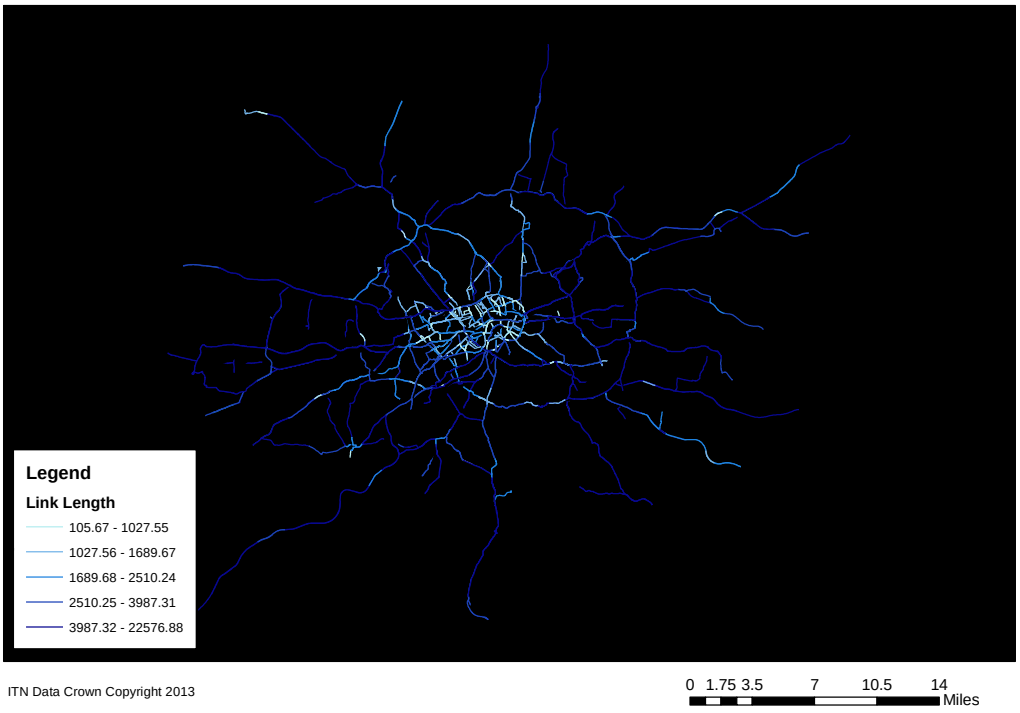


Figure 4.6 – Link lengths on the LCAP network (Symbology based on quantiles)

4.3 Quality of data on the LCAP network

The accuracy of forecasting systems depends on the quality of available data. The quality of data affects the certainty one can have in information derived from it. Uncertainty arises from measurements of objects or processes, which could consist of a range of values which may or may not include the true value (Shi et al., 2003). There are various sources of uncertainty in spatio-temporal data, which can be caused by errors, sampling design and missing data. In the following subsections, a thorough data quality analysis is carried out for the LCAP network. Data quality is analysed in the following respects: 1) Data accuracy and precision; 2) Data completeness; 3) the nature of missing data.

4.3.1 Factors affecting precision of LCAP data

To understand precision in the context of ANPR data collected on the LCAP network, it is necessary to understand what is being measured in each TT observation. Firstly, each vehicle that contributes to a single TT observation has a different departure and arrival time within the 5 minute interval of the measurement. Therefore, individual measurements are not actually taken under the same conditions. What is sought is a value of mean TT for that

period that is as close as possible to the mean TT experienced by drivers over that period. It is possible that individual vehicle TTs may have an upward or downward trend over a 5 minute interval, depending on whether traffic is building or decreasing. However, the time period is sufficiently short to assume that individual TTs are normally distributed within a given 5 minute period. Therefore, each individual TT is a sample from that distribution.

The precision of an aggregate TT observation depends on many unobserved factors. For example, a significant proportion of vehicles in urban areas are taxis. Taxis may stop at any time to pick up or drop off passengers, in which case their TT will be longer than other vehicles on the same link. This will increase the aggregated TT for that link. Similarly, the spatial layout of the sensor network affects precision. The network is link based, but it is not a closed system and the measured links represent only the best estimate of the likely route between two camera locations. It is entirely possible that a vehicle will take a different route between two cameras, thus distorting the LTT observation. If this type of deviation occurs on an individual vehicle basis, then it can be viewed as random error, and will affect the precision of an observation. Robinson and Polak (2006, p. 611) summarise the journey types that contribute to an aggregate TT observation as:

A. Vehicles that aim to traverse the link in a prompt manner.
B. Vehicles taking an alternative route between the start and end of a link.
C. Vehicles that have stopped en-route.
D. Vehicles that are not subject to normal traffic regulations, such as emergency vehicles.
E. Vehicles that do not aim to travel from the start of a link to the end of a link in the fastest manner possible.

It would be expected that the majority of journeys would be of type A. The influence of journey types B-E on aggregate TTs depends on the frequency of the data. The influence of journey type B may be influenced by the directness of the route. In the absence of distributional information on aggregate TTs in this thesis, observations with a higher frequency are considered to be more precise.

4.3.2 Factors affecting accuracy of LCAP data

There are many factors that could affect the accuracy of observations obtained on the LCAP network. Returning to the spatial layout of the sensor network, if vehicles systematically take a route between camera locations that is different from the assumed route, then travel

times will be systematically increased/decreased and estimates of dimensionless quantities such as speeds/UTTs will be systematically decreased/increased. This may occur if traffic along a route is diverted because of road works or some other planned or unplanned event. In this case, vehicles may still traverse a link between two camera locations, but the length of the route they take may be different. This poses problems in the context of forecasting, firstly because it changes the distribution of the observed travel times, meaning previously observed data may no longer be useful for model building, and secondly because it may indicate the presence of congestion where in fact there may be none. This second issue is a concern in spatio-temporal models because it may alter the correlation structure between adjacent locations on the network. In this thesis, it is assumed that the links are accurate representations of the route between two camera locations. However, it is acknowledged that the nature of the sensor network complicates the spatio-temporal relationship between links.

The sensor equipment itself can also be a source of inaccuracy. For example, incorrect synchronisation of the clocks at successive camera locations would result in estimates of speeds and UTTs again being systematically increased/decreased. These inaccuracies, if large, would result in incorrect estimates of link TTs. It is assumed in this thesis, however, that such inaccuracies would be too small to have a significant effect on TT observations.

4.3.3 Analysis of frequency of LCAP data

The frequency of an observation is the number of individual vehicles whose TTs are used to calculate it. A high frequency indicates a higher degree of certainty in the observation. Observations with low frequency are more susceptible to the influence of noise, and one can be less certain about their precision. The mean frequency of vehicles that contribute to a TT observation across the dataset is 4.23. Removing those observations for which no vehicles were observed (i.e. the data were missing) this rises to 7.51. Figure 4.7 is a histogram of the mean number of observations per link with missing values removed. The distribution appears to be a power law, meaning that a high proportion of links have a very low mean number of observations, while a low proportion have very high number of observations. Table 4.1 shows the deciles and minimum and maximum values with zero values removed. It can be seen that the number of vehicles contributing to travel time observations is rather low, with 80% of the data having below 7.76 vehicles contributing to each observation on average.

Table 4.1 – Deciles of mean number of observations per link with minimum and maximum values

Min	10%	20%	30%	40%	50%	60%	70%	80%	90%	Max
1.01	1.27	1.51	1.77	2.09	2.60	3.55	5.14	7.76	14.29	69.67

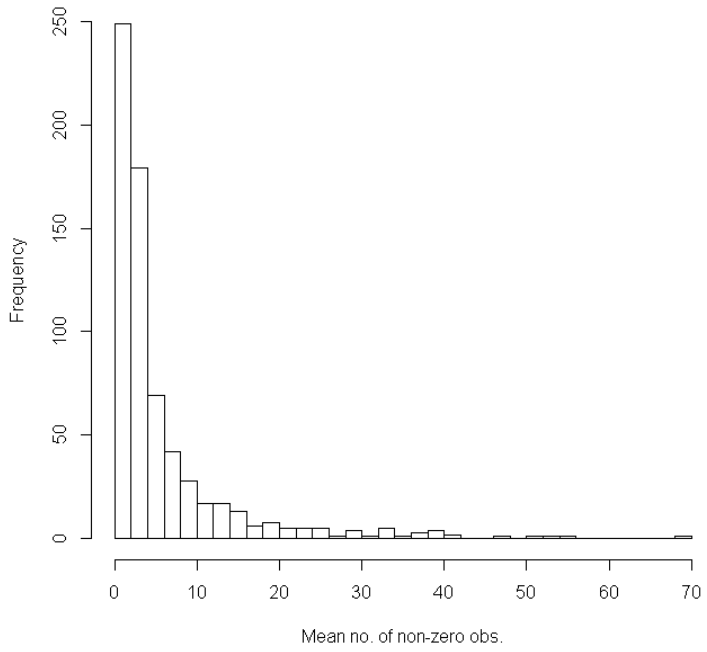


Figure 4.7 – Histogram of frequency of observation

The frequency of observation is related to the time of day at which the observation is made. This is shown in figure 4.8. All the weekdays are broadly similar; capture rates rise sharply from the overnight period before falling again in the middle of the day and subsequently rising after the evening peak period. There are two main explanations for this pattern. Firstly, in the early morning before the peak period begins, flow is high due to commuting traffic, but saturation point has not yet been reached so frequency is high. As the network becomes saturated, flow decreases causing a decrease in frequency. The second explanation is that, as density increases, the headway between vehicles decreases and fewer number plates can be successfully read at the start and end of links. It is likely that a combination of these two factors is responsible for the pattern. The weekend days are characterised by low capture rates in the early morning, which are caused by low traffic levels. Capture rates rise to high levels relative to the rest of the week from the middle period of the day into the

evening. The capture rates are probably higher than the other weekdays as there is less congestion and therefore more headway between vehicles, and less stationary traffic.

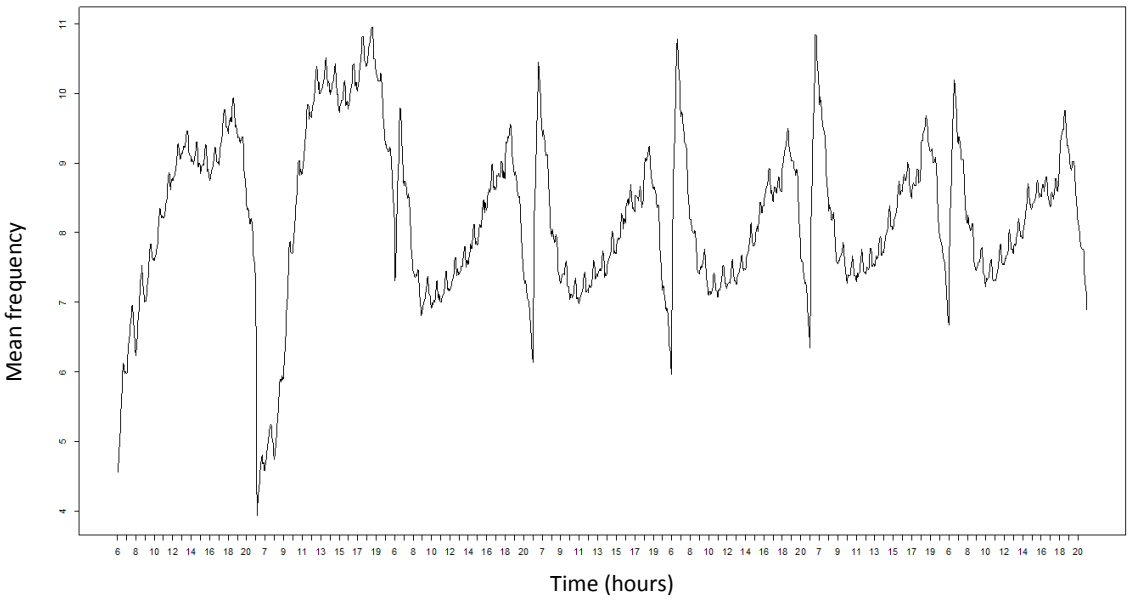


Figure 4.8 – Profile of mean frequency of observation across the week

Figure 4.9 shows the relationship between mean frequency and mean UTT across the test network. A striking pattern is evident, with a number of distinct features. Firstly, the relationship is very different between weekends and weekdays. On weekends, the relationship between frequency and travel time is broadly linear. As UTT increases, the frequency of observation increases. This reflects the positive relationship between UTT and flow. It is likely that the relationship remains linear because the network does not reach saturation point on weekends. On weekdays the relationship is nonlinear, but consistent between days. In the morning, as UTT begins to increase, the frequency of observation also increases up to around 6.35 am, at which point UTT continues to increase but frequency of observation decreases. The decrease continues to the AM peak period at around 8.30 am. Interestingly, the frequency of observation continues to decrease even as the UTT decreases after the AM peak period.

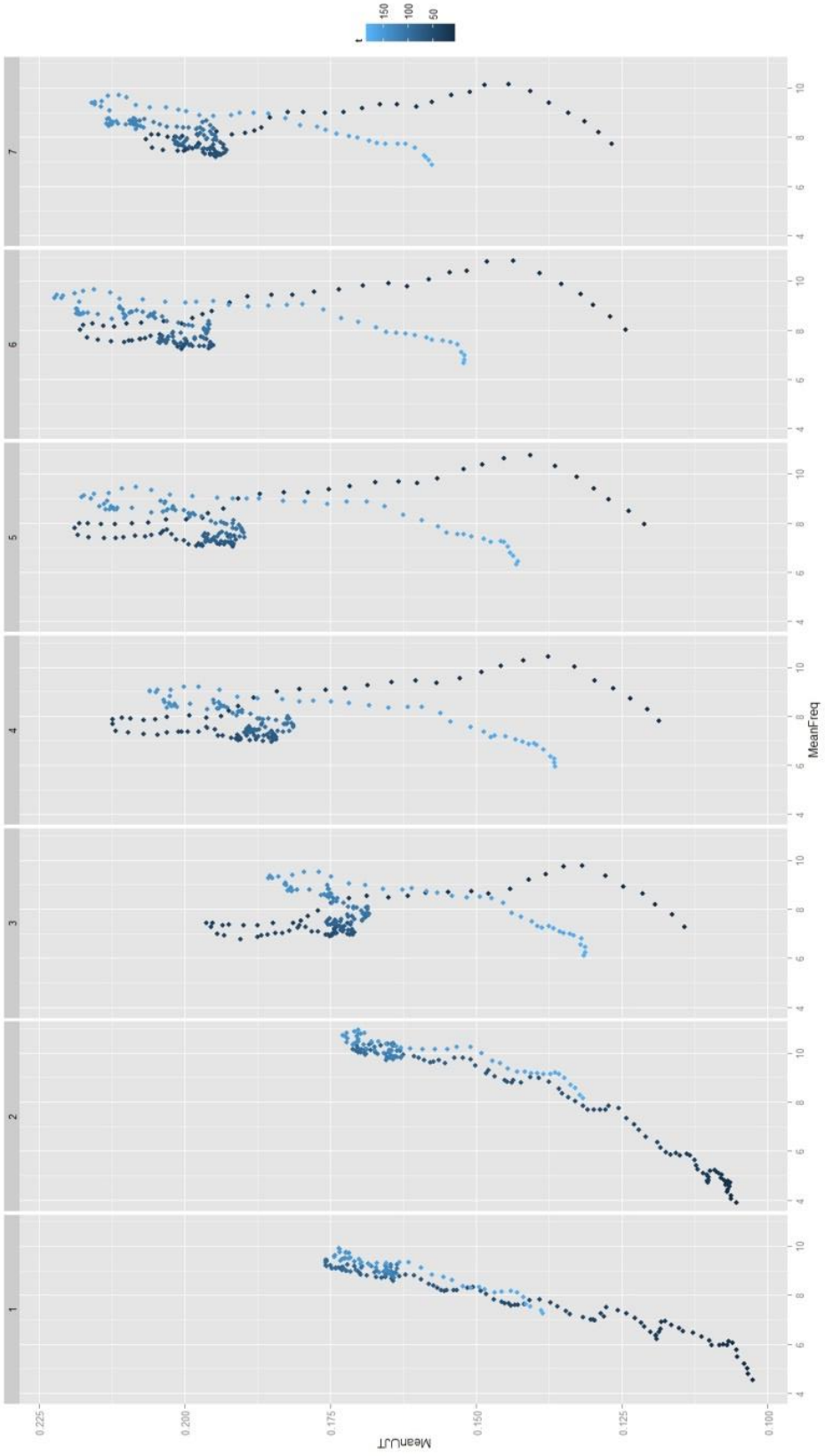


Figure 4.9 – Relationship between mean frequency and mean journey time across the test network. The colours represent time of day from 0 (6am, darkest blue) to 180 (9pm, lightest blue).

The findings here demonstrate that the likelihood of obtaining high frequency data is dependent on the time of day and the day of the week, which is in turn due to the cyclic variation in traffic states. Although not directly observable, it can be assumed that as traffic flow increases, frequency of observation increases up to a saturation point where it begins to decrease. The decrease in frequency can be attributed to one of two things; either there is insufficient headway between vehicles to record number plates at the start and end cameras of a link due to increased flow; or there are simply fewer vehicles completing the journey within the time interval due to congestion. The latter explanation could result from two situations; either vehicles divert from the route because of heavy traffic, or the speed of the traffic stream is lower so that fewer (if any) vehicles can exit the link within the time interval. The implications of this systematic variation in frequency vary in their severity depending on the time period. Low frequencies in the overnight period are not a major problem as the performance of the network at night is not a priority for transport agencies or road users. However, the decrease in frequency with increases in UTTs is more of a concern as it means that lower quality information is being collected at the times where high quality information is most needed; during congestion.

4.3.4 Analysis of completeness of LCAP Data

Data completeness, or missingness, is defined as the proportion of records for which there are observations. To examine the completeness of the LCAP dataset, it is first necessary to define the meaning of missing data in the current context. The function of an ANPR system is to measure individual travel times of vehicles traversing links. Each vehicle that traverses a link will either have its travel time recorded or not. Those vehicles for which the TT is not recorded can be seen as missing from the aggregate TT observation. However, because there is no data available to this study regarding traffic flows on the LCAP network, it is not possible to estimate the proportion of vehicles that are observed in an aggregate TT observation. Missing data in this context is taken to mean a 5 minute interval in which no vehicles are observed, and thus no aggregate TT is recorded. In TT data collected using ANPR, data can be missing for various reasons, some of which are related to the data collection system. These include:

A. Random error
B. Insufficient headway between vehicles for number plates to be read
C. Poor positioning of sensors
D. Faulty sensors

E. Poor light conditions
F. Vehicles not traversing the entire link
G. Stationary traffic caused by congestion.

When data are missing, they are replaced by TfL using an automatic process called patching, which is described in table 4.2.

Table 4.2 - Description of patch types used to replace missing data

Patch Type	No. of missing points	Interpolation Method
0	None	None
1	1	Average of adjacent observations.
2	2-6	Interpolated from adjacent observations.
3	> 6	Replaced with historical profile data (average of each time point).

The mean patch type for the network in the LCAP dataset is 1.19. 45.5% of observations have a patch type of 1 or above in the raw data. Of the 758 links in the network, 89 were inactive throughout the test period and are removed. Figure 4.10 is a histogram of missing data percentages per link. Overall, the frequency decreases as the missing rate increases, although there is a bump around 55% missing rate. Table 4.3 shows the percentage of each patch type in the data with the 89 inactive links removed. Patch type 3 accounts for the majority of missing data on the network, indicating that long term sensor failure (30 minutes or more) is a major issue on the LCAP network.

Table 4.3 – Percentage of each patch type after removal of inactive links

Patch Type	0	1	2	3
% of observations	61.76	4.99	9.74	23.51

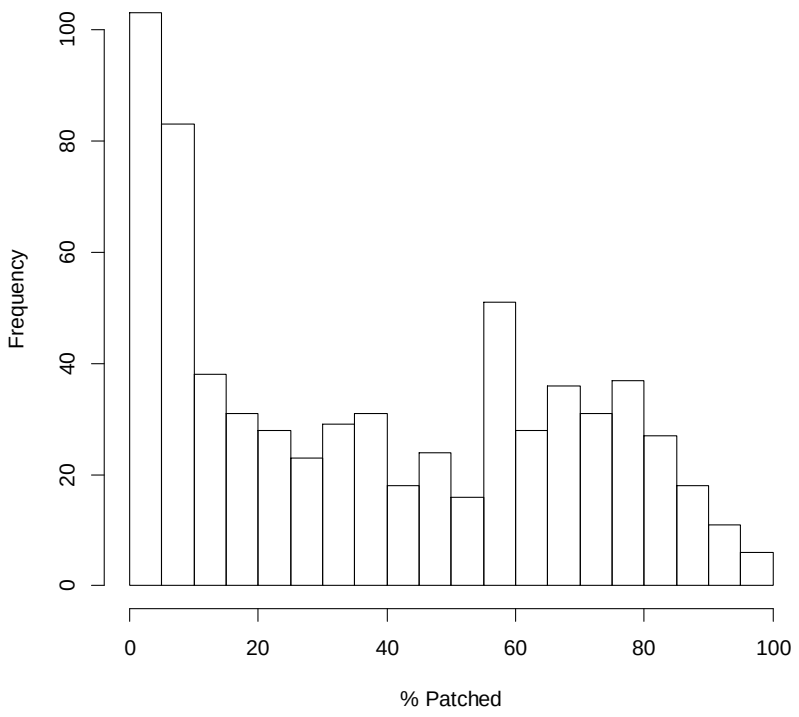


Figure 4.10 – Histogram of missing data percentages on the LCAP network

Figure 4.11 shows how the level of patching changes throughout the week. As with the mean frequency, there is a clear difference between the patching rate on Saturday and Sunday, compared with the rest of the week. There is an inverse relationship between the frequency of observation and the level of patching because as the frequency falls towards zero, there is a higher chance that no observation will be made.

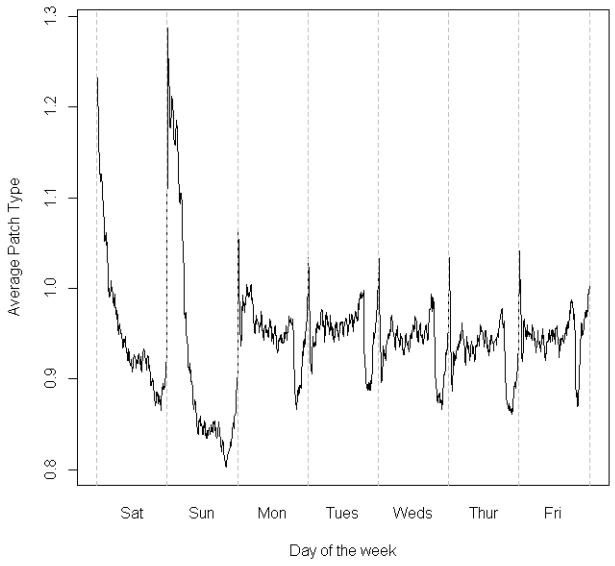


Figure 4.11 – Average Patch types across the week

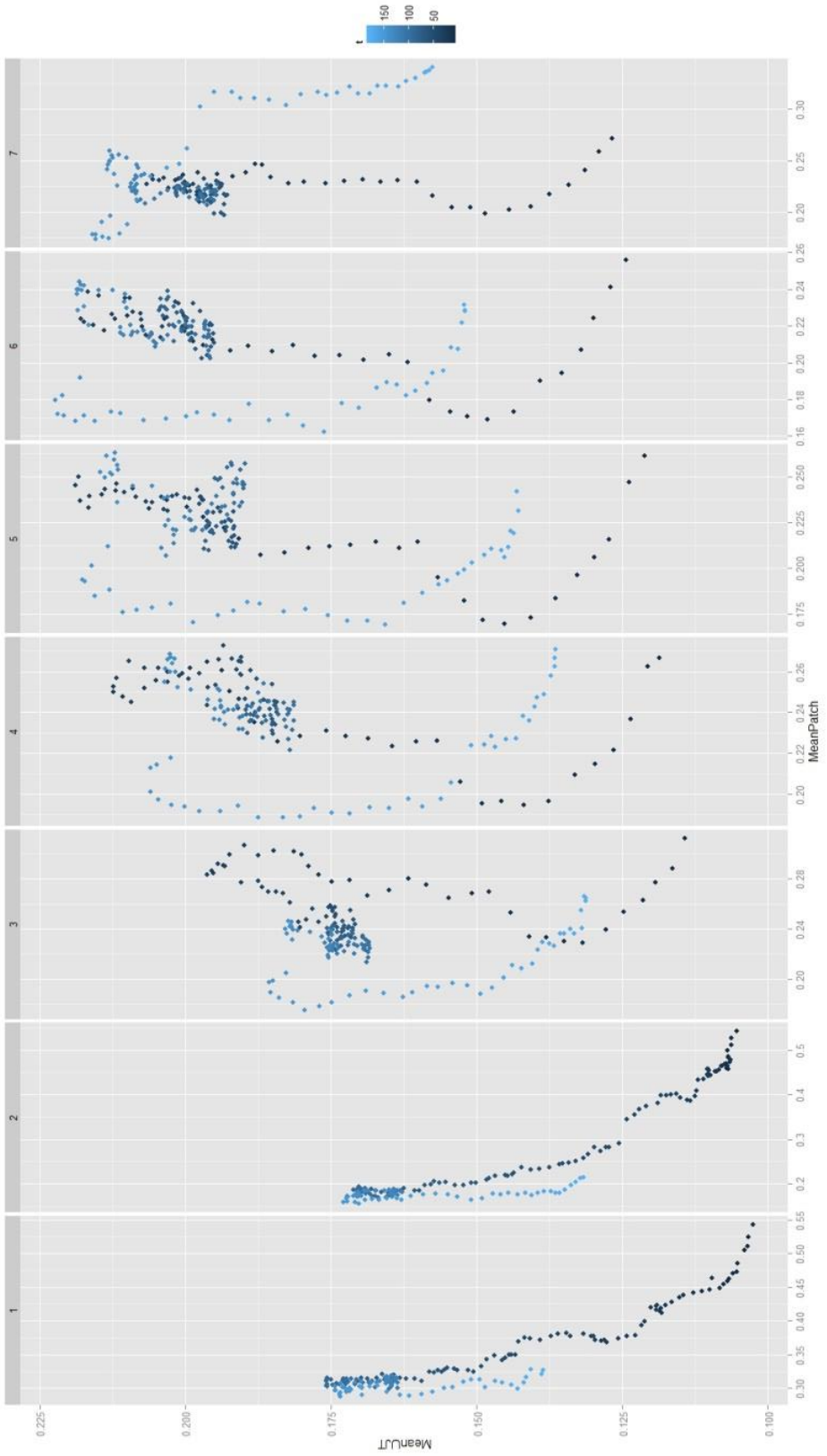


Figure 4.12 - Relationship between mean patch type and mean journey time across the test network. The colours represent time of day from 0 (6am) to 180 (9pm).

Figure 4.12 shows the relationship between mean patch type and mean UTT across the LCAP network. Like figures 4.8 and 4.11, the figure displays the inverse pattern of figure 4.9. On the weekends, the average patch type decreases as UTT increases because the increase in flow leads to a higher chance of vehicles being observed by the cameras. Again, the relationship is broadly linear. The maximum patch rates are higher on the weekend than on weekdays (approx. 0.55 compared with approx. 0.35) because there are fewer cars on the road early on weekend mornings and thus there is a higher chance that no vehicles will be detected over a given 5 minute period. Patch rates are high at 6 am on weekdays for the same reason. They decrease towards 6.30 am, before increasing again towards the peak period. On Monday to Thursday, the lowest patch rates are recorded in the evening after the PM peak has subsided. A different pattern is seen on Friday, where the highest patch rates are seen in the evening. This presumably reflects the different pattern of road usage on Fridays caused by night time leisure activities. This phenomenon is not, however, observed in the frequency data.

4.4 Data Cleaning and Preprocessing

In order to ensure unbiased results, both when carrying out exploratory analysis and when building predictive models, it is necessary to account for missing and corrupt data. Conveniently, the patching system provided by TfL provides a good indication of the value of a missing point. Data points of patch type 1 are isolated missing points which are interpolated from surrounding real observations before and after the failure. Therefore, they are likely to provide a reasonable estimate of the experienced travel time. Due to the longer period of missing data, patch type 2 is less likely to capture the conditions accurately, while patch type 3 provides no local temporal information whatsoever. In knowledge of this, the data cleaning process is as follows:

Data Removal

1. All links with 100% patched data are removed from the analysis as they are inactive sensors (89 links).
2. Those links that have a combined percentage of patch types 2 and 3 greater than or equal to 25% are removed (327 links).

Data Cleaning

3. Data of patch types 2 and 3 are removed from the results of all analyses. The procedure used to accomplish this differs depending on the type of analysis being carried out (i.e. autocorrelation analysis, forecasting) and so is described where necessary.
4. Data of patch type 1 remain as they are not expected to affect the results of the analyses. They are, however, removed from the calculations of error statistics in forecasting models. This avoids the removal of approximately 5% of the total data.

4.5 Summary of cleaned dataset

Figure 4.13 shows the final LCAP network that is used throughout this study. Compared with the full network shown in figure 4.3 a), the spatial coverage is significantly reduced in the peripheral areas, and to the West of Central London. However, the coverage within the CC Z is still good.

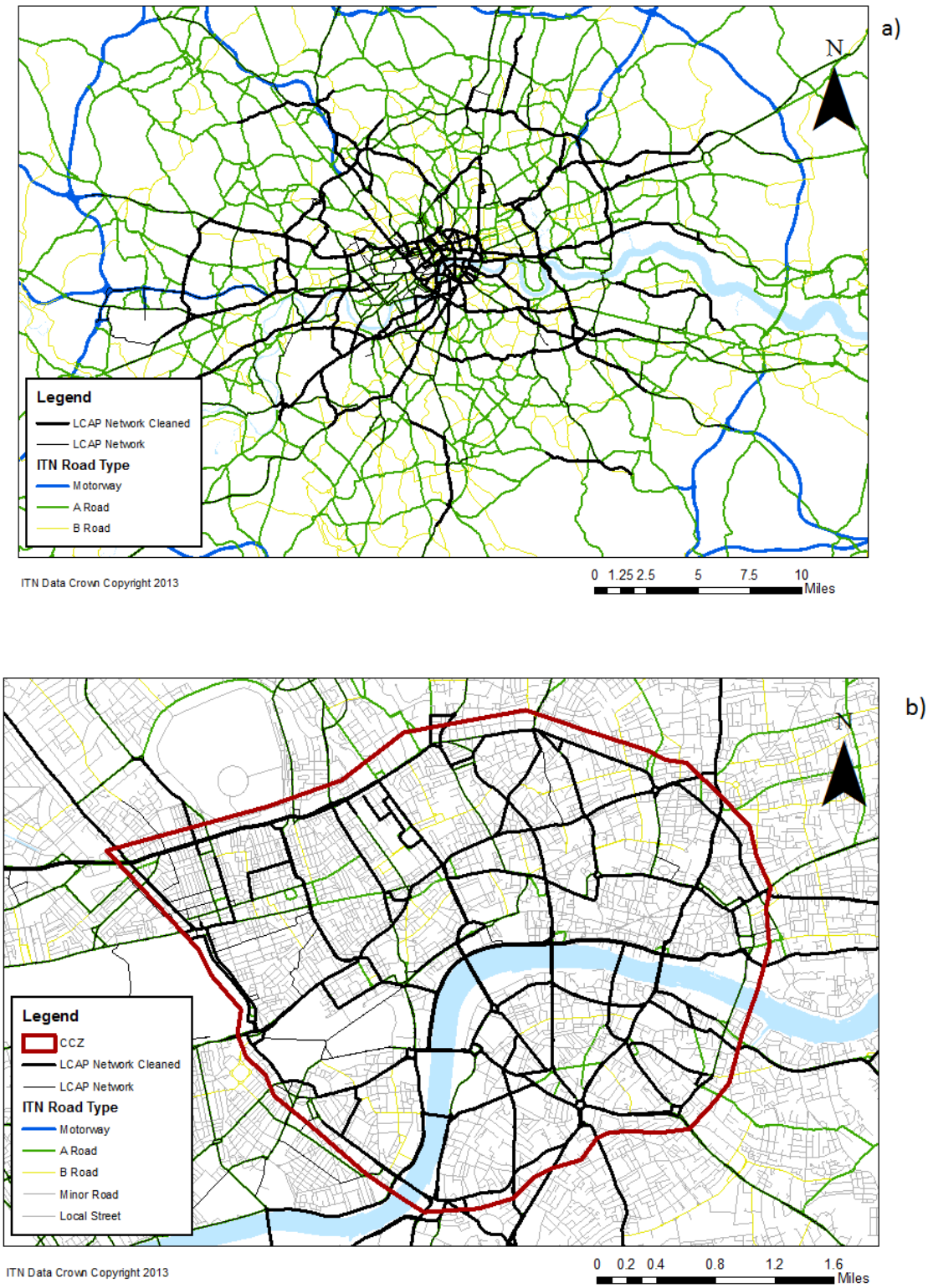


Figure 4.13 – Extent of the LCAP Network after data cleaning. a) The whole network. b) Within the congestion charging zone

4.5.1 Chapter Summary

In this chapter, the LCAP network has been introduced, which is a system of ANPR cameras that collect travel times on London's road network. The network is characterised by high levels of missing data and low frequency of observation. Furthermore, the network is spatially sparse. Within the CCZ, the LCAP broadly covers the network of *A* roads, thus accounting for the majority of traffic flows. In the periphery, its coverage is much less dense.

The LCAP network is characterised by low frequency of observation and high levels of missing data. According to the definitions of missing data described in section 3.7.1, these data are MNAR, and exhibit clear temporal patterns. There is a positive relationship between missing data rate and TT, and a negative relationship between frequency of observation and TT. The result of this is that the quality of data collection is lowest during congestion when accurate forecasts are most needed. This has significant implications from a modelling perspective: It implies that forecasting models that assume complete data will be insufficient for this network, and will exhibit poorest performance during peak periods where data are more likely to be missing. In turn, it motivates the use of models that are able to deal with the presence of missing data in real time.

It has also been revealed in this Chapter that a significant proportion of missing data (23.5%) is of patch type 3. Patch type 3 entails a missing sequence of greater than 6 points, which equates to 35 minutes or more. In the context of forecasting, a missing data point necessitates either a multi-step forecast in the univariate setting, or the incorporation of other data sources. When several data points are missing in sequence, accurate univariate forecasts become increasingly difficult to achieve. The implication of this is that univariate methods are likely to be insufficient for forecasting on the LCAP network in the presence of missing data, and a spatio-temporal approach is required.

In the following Chapter, a method is described for building an adjacency structure for the LCAP network. The method enables distances between sensor locations to be defined, thus allowing spatial structure to be included in forecasting models of the network.

Chapter 5 Modelling the LCAP Network

5.1 Chapter overview

In the previous Chapter, the LCAP network was introduced, and it was stated that a spatio-temporal model structure is necessary in order to deal with the large amounts of missing data. However, it was also noted that the LCAP network is spatially sparse. In this chapter, the representation of the LCAP network in a graphical structure is considered. The representation considers not only the adjacency defined by the link based sensor network, but also the adjacency defined by the underlying physical road network. This leads to a better representation of the spatial relationship between sensor locations. The Chapter is organised as follows: In section 5.2, the representation of the LCAP network as a graph is described. In section 5.3, a local autocorrelation analysis is carried out in order to explore the spatio-temporal dependency structure of the network, and some modelling implications are discussed. Finally, some requirements for space-time models of network data are stated in section 5.4, based on the analysis in this Chapter and the outcomes of Chapters 3 and 4.

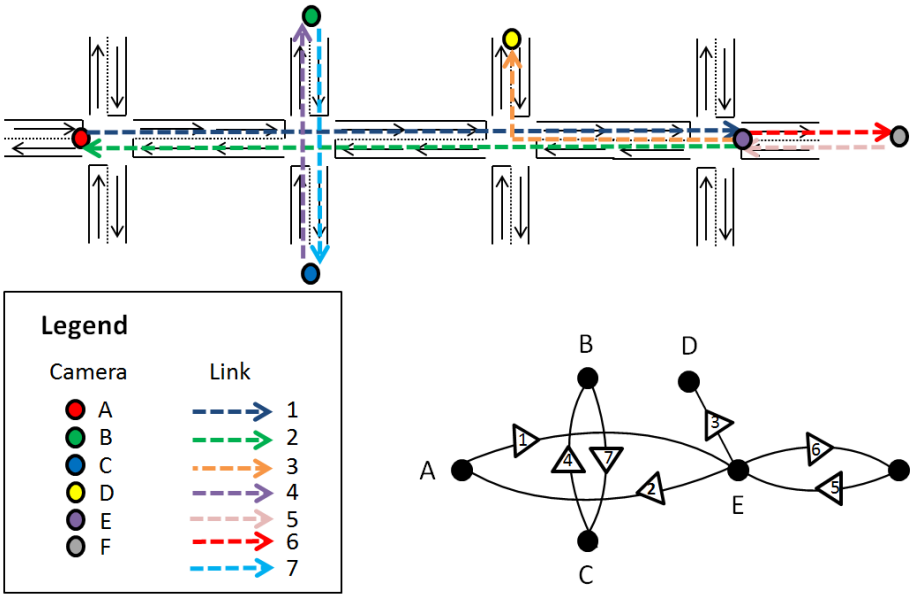
5.2 Modelling the LCAP network structure

In this section, the steps taken to represent the LCAP network in a mathematical model are described. The graphical structure of the LCAP network is the combination of two graphs, which are the sensor network structure and the physical road network structure. The LCAP exists upon, but is not fully coincident with the road network. Therefore, while it has an adjacency structure that is clearly defined by its links and camera locations, its structure does not account for all possible journeys between links. Therefore, a more complete representation is required. The section begins with a description of the adjacency structure of the LCAP network in section 5.2.1. In section 5.2.2 the connectivity structure of the physical road network is defined. Finally, in section 5.2.3, the two adjacency structures are combined to give a more realistic representation of the spatial relationship between LCAP links. This representation is used for forecasting of travel times under missing data in Chapter 7.

5.2.1 Adjacency structure of the LCAP network

The LCAP network is a link based sensor network that has a naturally defined adjacency structure. It is intuitive to view the ANPR camera locations as nodes and the road sections

linking cameras as edges. This representation mimics the physical structure of the sensor network and would be appropriate for monitoring systems in which point observations are recorded at the nodes, such as loop detector networks. However, in link based data it is on the road sections themselves where a quantity is observed. Of interest are the relationships between the observations recorded on the links, which are connected by the camera locations. Therefore, the camera locations form edges between links. Figure 5.1 shows a graphical representation of a typical sensor layout on the LCAP network.



c)

Link	1	2	3	4	5	6	7
1	-	1	1	-	-	1	-
2	1	-	-	-	-	-	-
3	-	-	-	-	-	-	-
4	-	-	-	-	-	-	1
5	-	1	1	-	-	1	-
6	-	-	-	-	1	-	-
7	-	-	-	1	-	-	-

d)

Link	1	2	3	4	5	6	7
Length (m)	1000	1000	500	300	250	250	300

e)

Link	1	2	3	4	5	6	7
1	-	1000	750	-	-	625	-
2	1000	-	-	-	-	-	-
3	-	-	-	-	-	-	-
4	-	-	-	-	-	-	300
5	-	625	375	-	-	250	-
6	-	-	-	-	250	-	-
7	-	-	-	300	-	-	-

Figure 5.1 – Adjacency structure of the sensor network: a) A sequence of intersections on which ANPR cameras are installed; b) the graph resulting from a) based on edge-edge adjacency; c) the adjacency matrix resulting from b); d) the link lengths; e) the spatial weight matrix resulting from b) and c).

In the figure, the camera locations are represented by the letters A-F and the links are represented by the numbers 1-7. Links can span several intersections, so traffic can enter and exit a link at locations other than its start and end points. The network is not planar, so links can cross other links without intersecting and their traffic can interact. Links can also partially cover the same section of road in the same direction. Examining figure 5.1 b), two graphs are formed from the layout in figure 5.1 a), one comprising cameras A, D, E and F, and one comprising cameras B and C. Although links 1 and 2 cross links 4 and 7, they are not considered to be adjacent in this representation[15]. Figure 5.1 c) shows the adjacency matrix formed from the graph in figure 5.1 b). The adjacency matrix is built based on *edge/edge adjacency* rather than *node/node* adjacency (Black, 1992). In effect, this means that the links are considered to be nodes. The adjacency matrix is directed according to the direction of the links, and is asymmetric. The same camera location can observe traffic in both directions in the traffic stream, which means that two links can be adjacent even if they flow in opposite directions along the same section of road. For example, link 4 is adjacent to link 7 and vice versa. Based on this structure, the graph $G_{L,S} = (L, S)$ is constructed to represent the LCAP, where $l = 1,2, \dots, L$ are the links and $s = 1,2, \dots, S$ are the sensor (camera) locations that connect them.

5.2.2 Spatial weight structure of the LCAP network

In addition to defining the spatial adjacency structure, spatial weights can be defined. The most straightforward weighting scheme to define is a distance based weighting scheme. The node/node representation provides a natural mechanism for capturing distances as edge weights in the adjacency matrix. However, in the edge/edge representation, the edges have zero length as they are points (camera locations) rather than lines, and it is the nodes (links) that have length. Setting the distance between links to be zero ignores the influence that link length has on the relationship between adjacent measurements. It would be reasonable to assume, however, that link length will have a non-negligible effect on the relationship between observations recorded on adjacent links because links that are closer to one another are more likely to exhibit similar traffic. The spatial heterogeneity in link length (shown in figure 4.5 in the previous Chapter) indicates that link length is an important concern on the LCAP network. Furthermore, ignoring link length precludes network distance as a proxy for the strength of spatial relationship between links.

[15] The complete graph of the LCAP network is fully connected, but the routes it defines do not always follow shortest paths, i.e. the two graphs would connect at a more distant location.

To measure distances between links on the LCAP network, the links (nodes) must be assigned to a physical point in space. This is accomplished by assigning the observations to the midpoints of the links. Thus, the distance $d_{i,j}$ between two adjacent links i and j is calculated as half the sum of their lengths:

$$d_{i,j} = \frac{1}{2}(len(l^i) + len(l^j))$$

(5.1)

Where $len(l^i)$ is the length of link i, $i, j = 1,2, \dots, L$. Using equation 5.1, the elements of a distance based spatial weight matrix $W_{L,S}$ for graph $G_{L,S}$ can be defined according to equation 5.2.

$$w_{i,j} = d_{i,j}\ iff\ (l^i, l^j) \in S$$

(5.2)

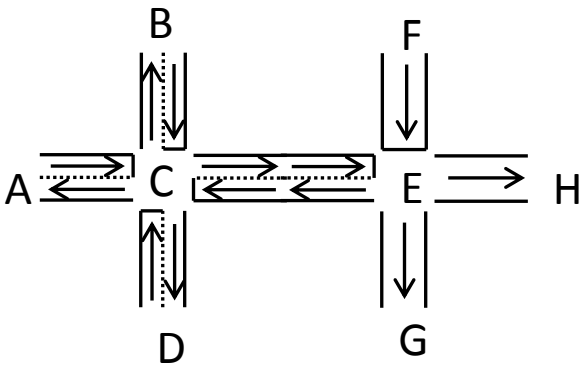
The weight matrix formed according to this scheme is shown in figure 4.13 e), based on the distances shown in figure 4.13 d). Although distance is used here, there are various other weighting schemes that could be applied. For example, weights could be based on expected travel times between locations on the network.

5.2.3 Adjacency structure of the physical road network

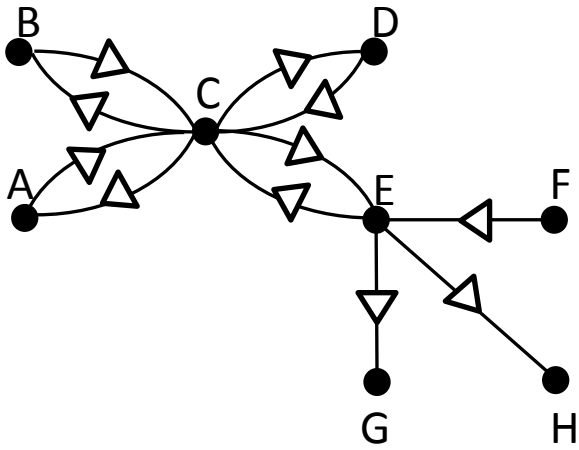
The physical road network is stored in Ordnance Survey's (OS) Mastermap Integrated Transportation Network (ITN) Layer, which is an intersection based, directed network. The ITN is available free of charge for academic use from EDINA's DIgimap service, at the following URL: *edina.ac.uk/digimap*. It has four elements: 1) A road network containing the geometry; 2) Road routing information (i.e. one way, mandatory turns etc.); 3) an upper layer containing information on bridges and overpasses etc., and; 4) urban paths (not required here). The road network (element 1) is represented in a GIS as two layers; a layer consisting of edges (road sections) and a layer consisting of nodes (intersections). Each edge has a start node and an end node associated with it, along with other attributes such as length and direction. The ITN representation is planar, meaning that a node exists wherever two roads cross. This is the case even when the roads do not intersect. For example, a node exists where a bridge crosses another road, or a tunnel runs under another road. The upper layer (element 3) defines these occurrences. All nodes that are included in element 3 are

removed, and the adjacent nodes on either side connected in order to prevent unrealistic routes.

a)



b)



c)

	A	B	C	D	E	F	G	H
A	-	-	1	-	-	-	-	-
B	-	-	1	-	-	-	-	-
C	1	1	-	1	1	-	-	-
D	-	-	1	-	-	-	-	-
E	-	-	1	-	-	-	1	1
F	-	-	-	-	1	-	-	-
G	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-

Figure 5.2 – Graphical representation of the ITN: a) a diagram of eight connected intersections; b) the resulting graph; c) the resulting adjacency matrix.

The function of the ITN in this thesis is to provide routing information between locations on the LCAP network. The representation of transport networks in routing models is

straightforward and has been researched extensively (see, for instance, Bell and Iida, 1997). The graphical representation defined by the ITN dataset is retained, which is termed the *node/node* representation. The ITN is defined as a directed graph $\mathrm{G_{N,E}} = (\mathrm{N}, \mathrm{E})$, where $\mathrm{n} = 1,2, \ldots, \mathrm{N}$ are the nodes and $\mathrm{e} = 1,2, \ldots, \mathrm{E}$ are the edges. This structure is shown diagrammatically in figure 5.2.

Figure 5.2 a) shows a typical grid based street layout. The letters A-H all represent intersections (nodes). The arrows represent the direction of flow. Road links (edges) with one flow direction are one way links. It is assumed that turns can be made in any direction at an intersection, provided it is permitted by the flow direction[16]. Figure 5.2 b) shows the directed graph resulting from figure 5.2 a). Bidirectional roads are represented by two directed edges, one flowing in each direction. Unidirectional roads are represented by a single directed edge. The resulting adjacency matrix is shown in figure 5.2 c). The matrix is asymmetric. Nonzero i, j entries indicate that traffic flows from node i to node j. The rows corresponding to nodes G and H have no nonzero entries because flows exit the network through them. Likewise, the column corresponding to node F has no nonzero entries because flows can only enter the network through it.

5.2.4 Spatial weight structure of the physical road network

The graph $\mathrm{G_{N,E}}$ is used for routing purposes. Therefore, along with the adjacency structure, it must also store the distances between nodes. $\mathrm{G_{N,E}}$ is, therefore, a weighted graph, where the edge weights are the physical distances between nodes. Based on this structure, an $N * N$ spatial weight matrix $\mathbf{W}^{\mathrm{N,E}}$ is constructed according to equation 5.2 above. In this case, the distance $d_{i,j}$ is equal to the length of the edge (road segment) that connects nodes i and j. The ITN dataset for London used in this work consists of *S*=519,699 nodes in total. Therefore, the matrix $\mathbf{W}^{\mathrm{N,E}}$ is stored in sparse form using the SparseM library in R statistical package (Koenker and Ng, 2013). $\mathbf{W}^{\mathrm{N,E}}$ can be used to calculate routes between pairs of nodes on the ITN network, and their corresponding lengths. This is accomplished by converting it to a graph object using the igraph library and using the inbuilt shortest path methods (Csardi and Nepusz, 2006). Dijkstra's algorithm is the method used in the package (Dijkstra, 1959).

[16] Restricted turns are accounted for in the ITN adjacency structure, but are not shown here for clarity of presentation.

5.2.5 Improving the LCAP network representation

In previous work on this network (Cheng et al., 2011; Haworth and Cheng, 2012; Cheng et al, 2013), the adjacency structure of the sensor network as defined by $G_{L,S}$ was used to define the spatial relationship between sensor locations. This restricts the spatio-temporal neighbourhood of a link to include only those links that share its start or end cameras. Such links are connected in direct sequence in an upstream or downstream direction, or in parallel in the opposite direction. This representation is sufficient when data have high spatial resolution. However, when dealing with traffic links of long length, the nearest spatio-temporal information can be spatially distant. For example, the midpoints of two adjacent links of length 1km are 1km apart. In an urban setting, there is scope for a large amount of variation in traffic conditions over a 1km stretch of road, particularly if there are many intersections along its length. However, if two links cross near their midpoints, their traffic will interact directly. On the LCAP network, links also often incorporate turns, meaning that the path between sensor locations may not be certain. The LCAP adjacency scheme defined in $G_{L,S}$ has particularly severe implications during sensor failure because a single camera failure can result in the loss of data from multiple links (see Chapter 6). Therefore, the step is taken to fuse the structure of the road network with that of the sensor network in order to develop a more accurate representation of the true spatial connection between sensor locations.

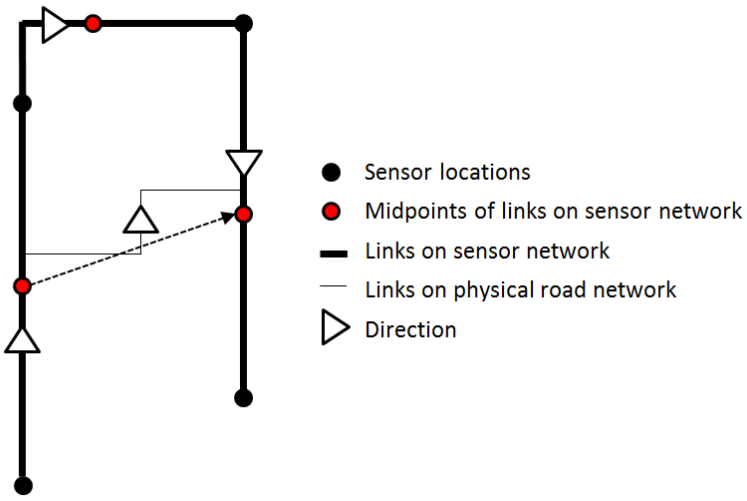


Figure 5.3 – Distances measured between sensor locations using Euclidean distance, LCAP (sensor) network distance and road network distance

Figure 5.3 shows a simplified representation of the effect that the connectivity structure of the sensor network has on the measured distances between locations. The thick black line represents a path along the LCAP network, the thin black line represents a path along the physical road network, and the dotted black line represents the Euclidean distance. When travelling between the two midpoints connected by the dotted line, the LCAP distance exaggerates what may be the true network distance, whereas the Euclidean distance underestimates it. Although the Euclidean distance may provide a measure that is close to the network distance in some cases, the presence of physical barriers such as train lines, rivers and green spaces can lead to large underestimations in some cases.

5.2.5.1 Defining the adjacency between the physical and sensor networks

Joining the sensor network to the physical network involves joining two separate graphs, $G_{L,S}$ and $G_{N,E}$. The graphs exist in the same geographic space, but their structures are not coincident. To join the two graphs, the midpoints of the LCAP links are again used to represent their geographic location. However, instead of calculating distances along the LCAP only, distances between LCAP links are calculated along the physical road network. Clearly, there may be a large number of possible routes between sensor locations, particularly if they are separated by large distances. Therefore, for simplicity, the shortest path distance is used, which is calculated using Dijkstra's algorithm. For those LCAP links that are adjacent in terms of the sensor network, the shortest path may be the same as the LCAP path and their separation distance will remain the same. Shortest path has received criticism in the literature for its simplification of human route selection behavior. However, it is appropriate in this case because the relationships of interest are those between locations that are very close in space, and the shortest path is more likely to be the true path over shorter distances as there are fewer alternatives.

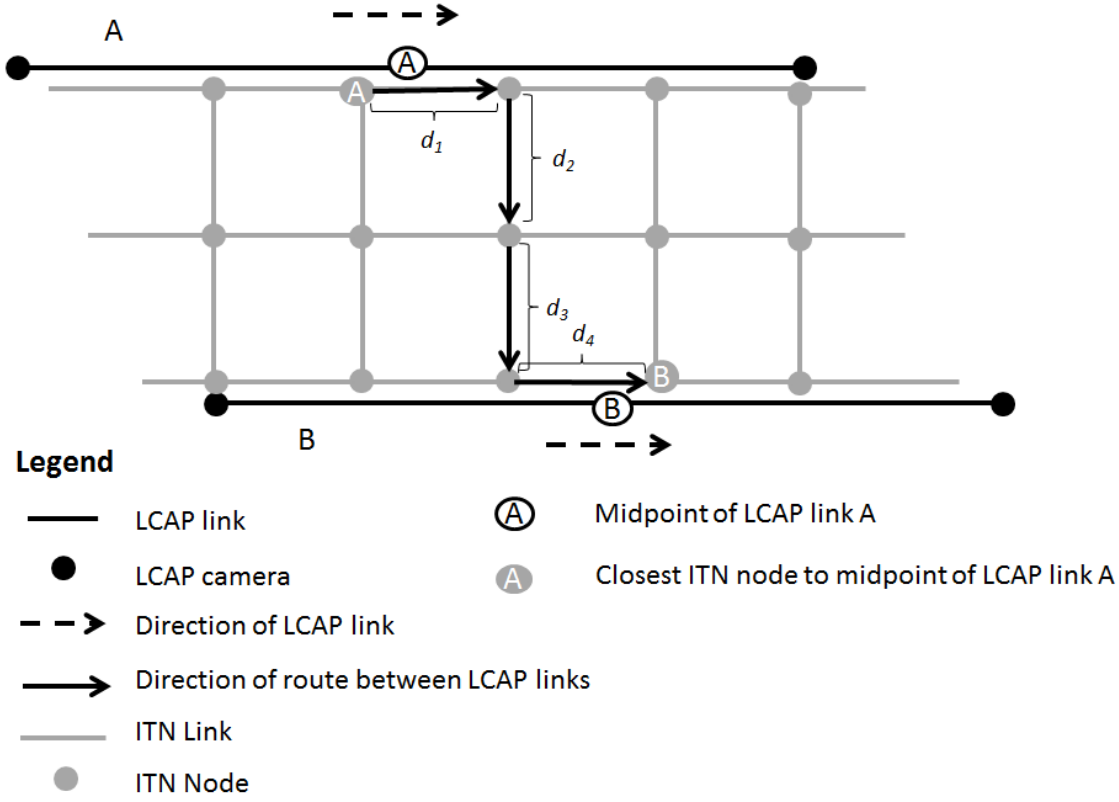


Figure 5.4 – Illustration of the process of connecting the sensor (LCAP) network to the physical (ITN) network

To relate the location of the LCAP midpoints to the ITN network, the closest ITN node to the midpoint of each LCAP link that falls on the LCAP link in question is used. A new graph $\mathrm{G}_{\mathrm{L,E}}$ is constructed, that combines the nodes of graph $\mathrm{G}_{\mathrm{L,S}}$ with the edges of graph $\mathrm{G}_{\mathrm{N,E}}$. The nodes N of $\mathrm{G}_{\mathrm{N,E}}$ disappear and the edges E become routes between the nodes L of $\mathrm{G}_{\mathrm{L,E}}$. The lengths of these routes provide the edge weights, thus $\mathrm{G}_{\mathrm{L,E}}$ is a directed, weighted graph.

The process is shown diagrammatically in figure 5.4. The midpoints of LCAP links A and B (nodes of $\mathrm{G}_{\mathrm{L,S}}$) are assigned to the nearest nodes on the ITN network (nodes of $\mathrm{G}_{\mathrm{N,E}}$). The shortest path is then found between these nodes, shown as grey circles with the letters A and B in them[17]. The total length of the route is calculated as the sum of the lengths of the ITN edges (edges of $\mathrm{G}_{\mathrm{N,E}}$) it is comprised of. In the figure, this equals $d_1 + d_2 + d_3 + d_4$. The igraph library in R statistical package is used to calculate the shortest paths, and the i, j

[17] Note that in this simple grid structure, there may be more than one route of equal length between two midpoints (in this case three). This is very uncommon in practice, especially in road networks like that of London, which does not have a regular grid structure.

elements of $\mathbf{W}_{\mathrm{L,E}}$ contain the shortest path distance between links i and j. $\mathbf{W}_{\mathrm{L,E}}$ is in fact a distance matrix that does not have nonzero entries.

5.3 Local autocorrelation analysis of the LCAP network

In this section, an autocorrelation analysis is carried out to assess the level of temporal and spatio-temporal autocorrelation present in UTTs recorded on the LCAP network. The aim of this analysis is to assess the type of model that is likely to be successful in forecasting TTs on the network. This section is organised as follows: In section 5.3.1, the method for measuring temporal and spatio-temporal autocorrelation, the cross-correlation function (CCF) is described. Following this, in section 5.3.2, the interpretation of the values of the CCF is considered. In section 5.3.3 the purely temporal autocorrelation in the TTs is analysed. Following this, the spatio-temporal cross-correlation (CC) is analysed in section 5.3.4. Finally, the modelling implications of the space-time autocorrelation structure are given in section 5.3.5.

5.3.1 Measuring local space-time autocorrelation

To examine the local autocorrelation properties of the TTs collected on the LCAP network, the cross-correlation (CC) function (CCF) is used. The CCF (see, for example, Box et al., 1994) treats two time series as a bivariate stochastic process and measures the cross covariance coefficients between each series at specified lags. Intuitively, it provides a measure of the similarity between two time series. Given two time series X and Y, the CC $\rho_{xy}(k)$ at lag k is given as:

$$\rho_{xy}(k) = \frac{E[(x_t - \mu_x)(y_{t+k} - \mu_y)]}{\sigma_x \sigma_y} \qquad k = 0, \pm 1, \pm 2, \pm \cdots \tag{5.3}$$

Where σ_x and σ_y are the standard deviations, and μ_x and μ_y the means, of series X and Y respectively. The CCF is used to measure CCs in both directions. The temporal lag at which the CCF peaks can be used to determine a transfer function between two series, provided the peak does not occur at lag zero. The temporal autocorrelation function (ACF) is a special case of the CCF in which series X and Y are the same series. The CCF is ideally suited to measuring local space-time autocorrelation of network data because it enables the calculation of pairwise CCs. A pairwise CC function is more informative than a

neighbourhood based CC function in the network case as averaging correlation in the spatial neighbourhood of a road link will mask the relative contribution of each link to the value of CC. The CCF allows the statistical relationship between the TTs on pairs of road links to be measured.

5.3.2 Statistical significance and interpretation of cross-correlation

To test whether the CC between two series is significant, confidence intervals are applied. The variance of the CC coefficient under the null hypothesis of zero correlation is approximately 1/n, where n is the length of the series. Approximate critical values at the 95% confidence interval are $\pm 2/\sqrt{n}$. With long series, this typically results in very low actual values of CC being identified as significant. For example, with a series of $n = 100{,}000$, the confidence intervals are evaluated as $\pm \frac{2}{\sqrt{100000}} = 0.006$. Therefore, statistically significant cross-correlation may not actually be informative from a modelling perspective. In fact, the primary function of the statistical test in the literature is often to identify correlation that is not significant, in order to satisfy the assumptions of linear time series and space-time series models. In the context of spatial autocorrelation, Griffith (2009) argues that autocorrelation values can be separated into 4 categories: 1) weak autocorrelation (0.25-0.5); 2) moderate autocorrelation (0.5-0.7); 3) strong autocorrelation (0.7-0.9) and; 4) marked autocorrelation (0.9-1.0). These are arbitrary categories, but they give context to the results of an autocorrelation analysis and are used here to aid the analysis.

5.3.3 Temporal autocorrelation in LCAP travel times

The first stage of the analysis involves the examination of temporal autocorrelation. The ACF is used to measure the temporal autocorrelation in each of the 342 links in the cleaned dataset that is described in section 4.5. The tests are carried out using the *stats* library in R statistical package (R Core Team, 2013). The patched values are removed from the analysis and are not included in the estimates of autocorrelation.

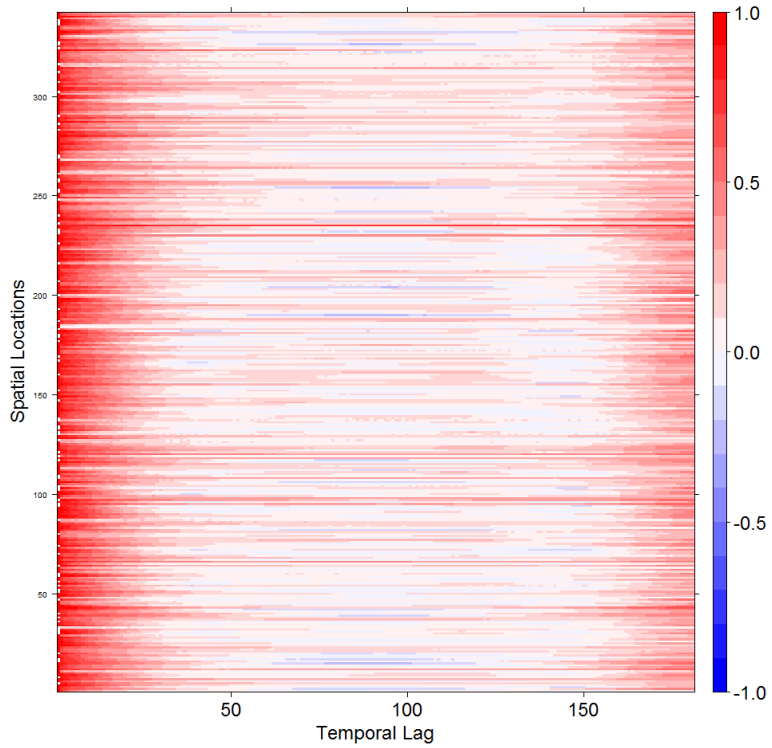


Figure 5.5 – Temporal autocorrelation in UTT for the 342 test links at the 5 minute aggregation level.

Figure 5.5 shows the ACF for each link in the test network. The red, white and blue colours indicate positive, zero, and negative autocorrelation respectively. The maximum temporal lag shown on the figure is $k = 180$, which equates to a lag of one day. The data is highly temporally autocorrelated. Strong, positive nonseasonal autocorrelation is present for the first 20-30 temporal lags in the majority of links. The value of the coefficient falls towards zero, and is sometimes weakly negative at a temporal lag of around $k = 90$. This lag equates to half a day of separation. The level of autocorrelation increases from this point, and becomes strongly positive at a lag of $k = 180$. This pattern is explained by the daily, deterministic variation in travel times caused by peak rush hour traffic, which occurs at (roughly) the same time(s) each day.

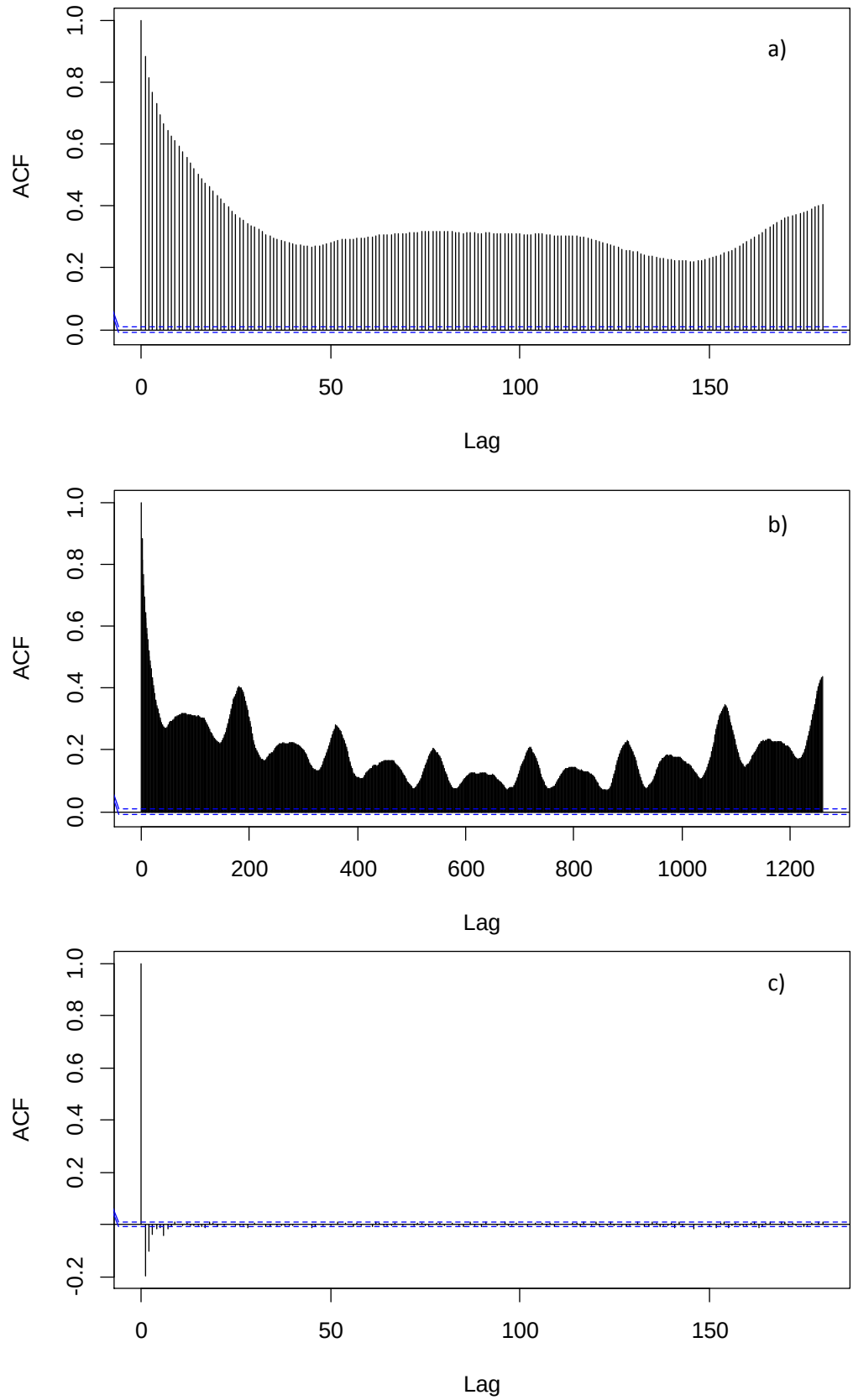


Figure 5.6 – Seasonal autocorrelation in LCAP data, a) Daily seasonal pattern, b) weekly seasonal pattern, c) ACF after first differencing (blue lines are approximate 95% confidence intervals).

Figures 5.6 a) and b) show the temporal ACF plots of a single link (link 417) with 180 lags (one day) and 1260 lags (one week) shown respectively. In these plots, both daily and weekly seasonal autocorrelation is evident.

5.3.4 Spatio-temporal cross-correlation in LCAP travel times

In this section, a cross-correlation analysis is carried out to assess the level of spatio-temporal autocorrelation in the data. Before proceeding, it is important to acknowledge that the presence of strong temporal autocorrelation that was identified in the previous section artificially inflates estimates of cross-correlation between time series, leading to incorrect inferences being drawn (Olden and Neff, 2001).

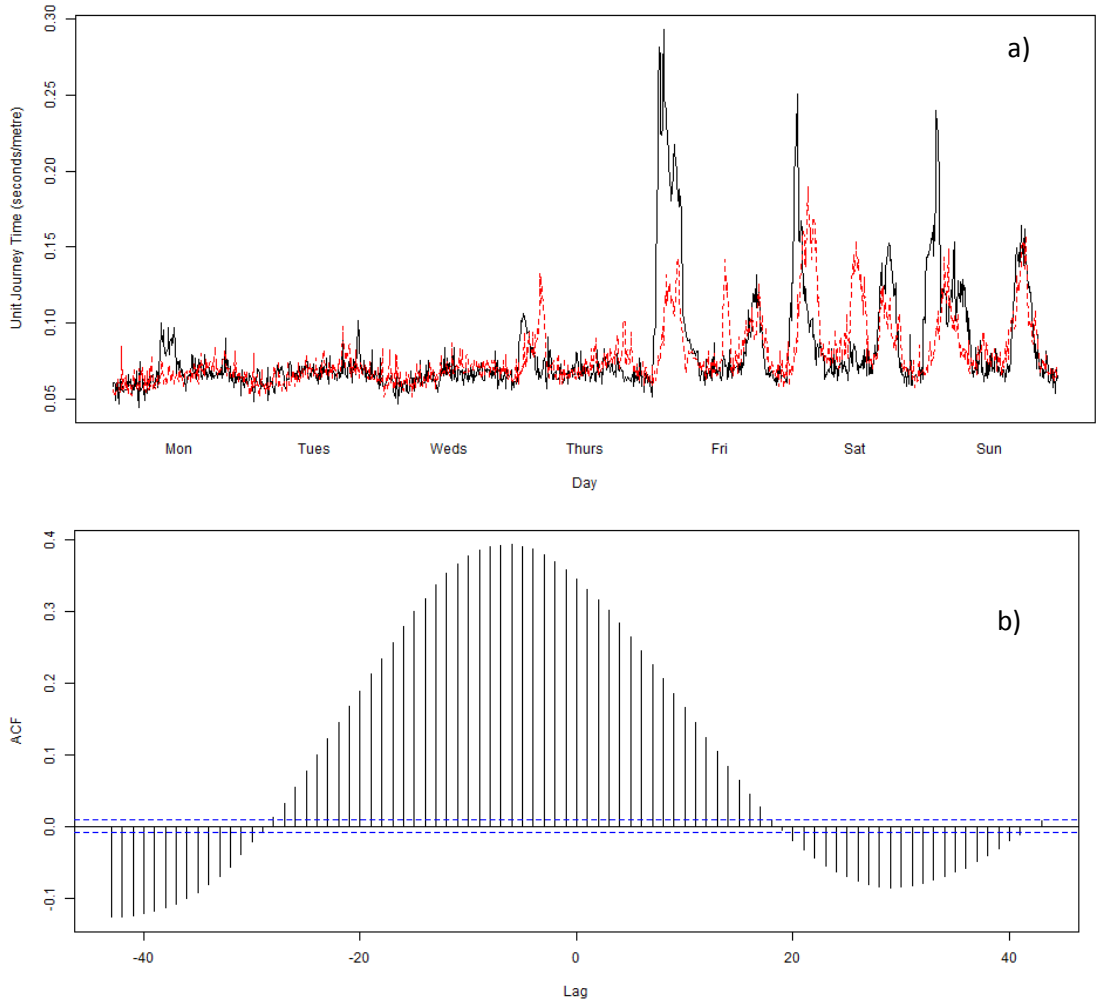


Figure 5.7 – Illustration of the danger of spurious correlation a) Two series of travel times collected at links on opposite sides of the network. b) The calculated cross correlation function between the two series, which shows significant correlation with slow decay and a seasonal trend (blue lines are approximate 95% confidence intervals).

As an example, figure 5.7 a) shows the UTT series of two links on opposite sides of the network. On first glance, they appear to be correlated with one another. This is confirmed on examination of the CCF, which shows strong, significant, positive CC between the two series. However, one would not expect there to be a causal relationship between the UTT values observed on these two links as they are approximately 21.5km apart (straight line distance). Their values are correlated because they both demonstrate similar AM and PM peak patterns. This phenomenon is known as spurious cross-correlation. Any inferences made based on this type of correlation would be invalid, and its presence makes it difficult to discern between causal and casual correlations.

5.3.4.1 Data Transformation

In the presence of temporal autocorrelation, the data must be transformed to stationarity in order to measure the true cross correlation between spatial locations[18]. The standard method for removing autocorrelation from time series is through differencing. The first order difference of a series is defined as (Kendall and Ord, 1990, p. 37):

$$\nabla y_t = y_t - y_{t-1} \tag{5.4}$$

The second order difference is then defined as:

$$\begin{aligned} \nabla^2 y_t &= \nabla(\nabla y_t) \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned} \tag{5.5}$$

And so on. Differencing involves subtracting the current value of the series from a previous value of the series at a specified temporal lag, producing a new series of differences that describes how much the value of the series changed during that time lag. Sometimes seasonal differences are used, with a seasonal lag S:

[18] Note that the transformations described here are not necessary for the operation of the models described in chapters 6 and 7, but are carried out here as they are necessary for the results of the cross-correlation analysis to be valid.

$$\nabla_S y_t = y_t - y_{t-S}$$

(5.6)

It is common practice to first attempt a first order nonseasonal difference at temporal lag 1 as this is often sufficient to remove autocorrelation from time series, and this is the approach taken here. Figure 5.8 a) shows the values of the ACF after first order differencing, at the 5 minute interval.

A first difference is sufficient to remove the seasonal and nonseasonal autocorrelation in the majority of links Figure 5.6 c) shows the resulting ACF plot up to temporal lag 180 on link 417. A small number of links still exhibit some weak positive autocorrelation at lags approaching 180, but in the general pattern is that a first difference is sufficient to remove the temporal autocorrelation from the data. Figure 5.8 b) shows the same data, but with only lags 1-10 displayed[19]. The majority of links exhibit strong negative autocorrelation at lag 1. The interpretation of negative autocorrelation at lag one in a differenced series is that the series tends to switch from a positive difference to a negative difference at each time step. That is, if the difference between the value of the series at time $t-1$ and t is negative, then difference between the value of the series at time t and $t+1$ is likely to be positive.

[19] Lag zero is omitted as it is always equal to 1.

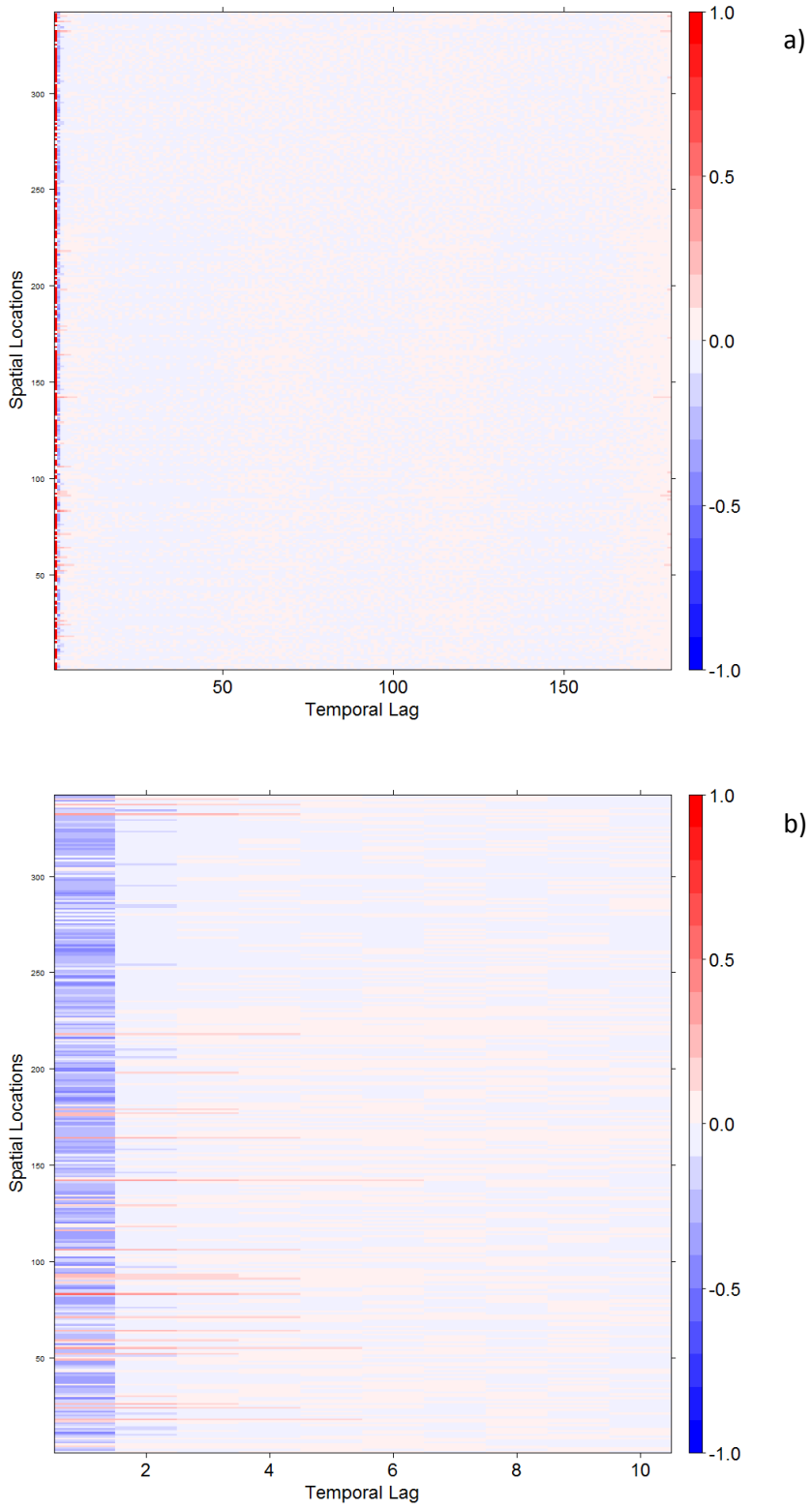


Figure 5.8 – ACF after first order differencing a) 180 lags shown. b) First 10 lags shown (excluding lag zero).

5.3.4.2 Results of spatio-temporal cross-correlation analysis

Pairwise CCs are calculated between the differenced UTTs at all links. The calculated coefficients are plotted against the measure of network distance defined in the graph $G_{L,E}$. Figure 5.9 a) shows the relationship between cross-correlation and ranked distance based on $G_{L,E}$. Each row of the plot represents a link and the columns are ranked by increasing distance from the given link[20]. Only the 20 closest links are shown as the cross-correlations become broadly insignificant at higher lags. The presence of deep red coloured cells on the left of the plots indicates that significant cross-correlation exists between links at short separation distances. There is a pattern of distance decay in the strength of cross-correlation. For comparison, the distance measure defined by $G_{L,S}$ is shown in figure 5.9 b). $G_{L,S}$ distorts the distances between locations, resulting in an inaccurate ordering of the network space. This means that significant, positive cross-correlations are observed at higher spatial lags.

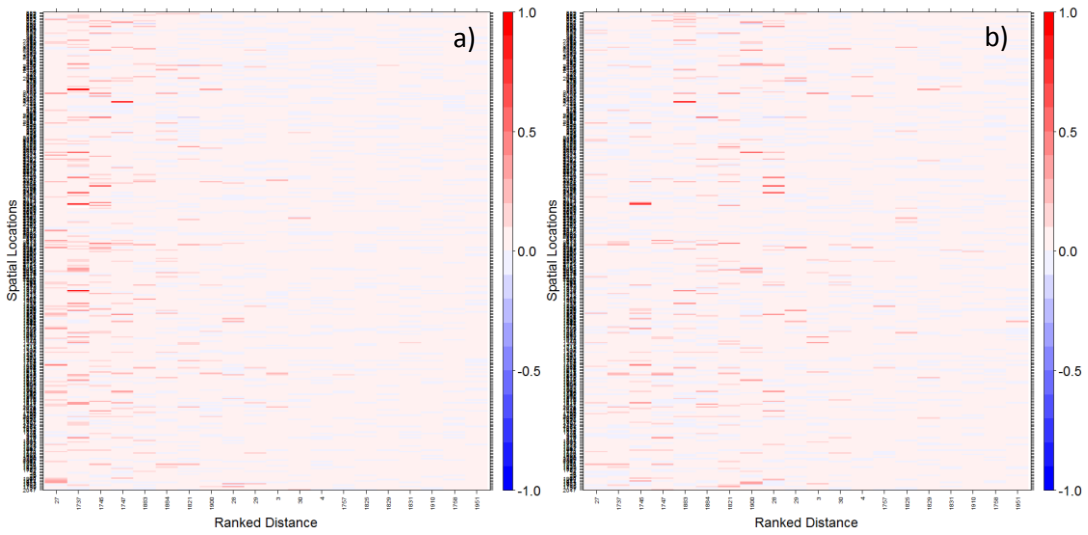


Figure 5.9 – Relationship between distance and CC at temporal lag zero based on a) $G_{L,E}$; b) $G_{L,S}$

5.3.4.3 Spatial pattern of cross-correlation

Although the general pattern is one of distance decay, there is evidence of spatial heterogeneity in the cross-correlation between locations. Firstly, some links are uncorrelated with any of their neighbours, suggesting that the propagation of traffic

[20] A ranked distance is used rather than the actual distance in order to retain the graphical representation.

conditions cannot be observed from the data in these cases. Secondly, those links that display significant cross-correlation with neighbouring links are not always most correlated with those links that are nearest in terms of network distance. This suggests that network distance alone does not explain the spatial relationship between links.

Figure 5.10 is a choropleth map showing the average cross-correlation between each link and its first five nearest neighbours according to $\boldsymbol{G_{L,E}}$[21]. Blue colours represent negative cross-correlation (significant and insignificant), white colours indicate insignificant positive cross-correlation at the 95% confidence interval, assuming a series length of 49,320. The red colours indicate varying strengths of statistically significant positive cross-correlation at the 95% confidence interval. The general pattern is one of statistically significant, but weak cross-correlation. However, the pattern is not uniform across space. Localised areas exist where the strength of correlation is higher, for example, the outer eastern area of the network where longer links meet at major junctions.

[21] An average of 5 neighbours is shown rather than just the nearest neighbour correlation because the peak cross-correlation is not always with the nearest neighbour.

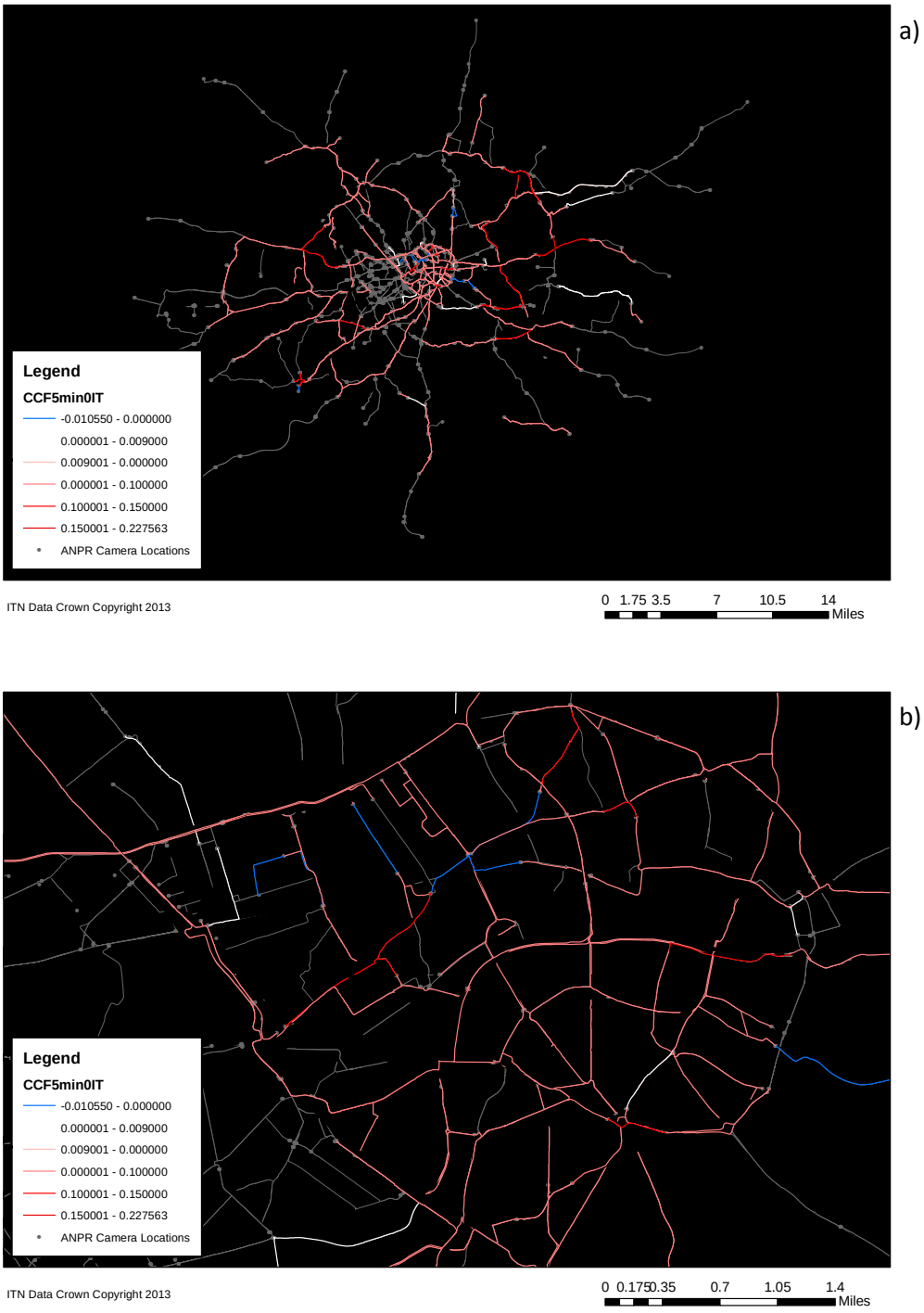


Figure 5.10 – Average cross-correlation between each link and its first 5 nearest spatial neighbours using the ITN distance measure defined by $G_{L,E}$ on: a) the whole network, and; b) within the CCZ

Figure 5.10 b) shows the same data focused on the CCZ. Here, the pattern of weak, positive correlation is repeated, but there are some areas with stronger correlation, and some with insignificant or negative correlation.

5.3.4.4 Strength of cross-correlation

The maximum value of CC observed on the network at temporal lag zero is 0.705. According to the categories of Griffith (2009), this constitutes strong positive autocorrelation, indicating the presence of spatio-temporal dependence between the links in question. However, the sample mean cross-correlation at temporal lag zero and spatial lags 1 and 2 combined is 0.059. Figure 5.11 is a histogram of the CCF values at these spatial and temporal lags.

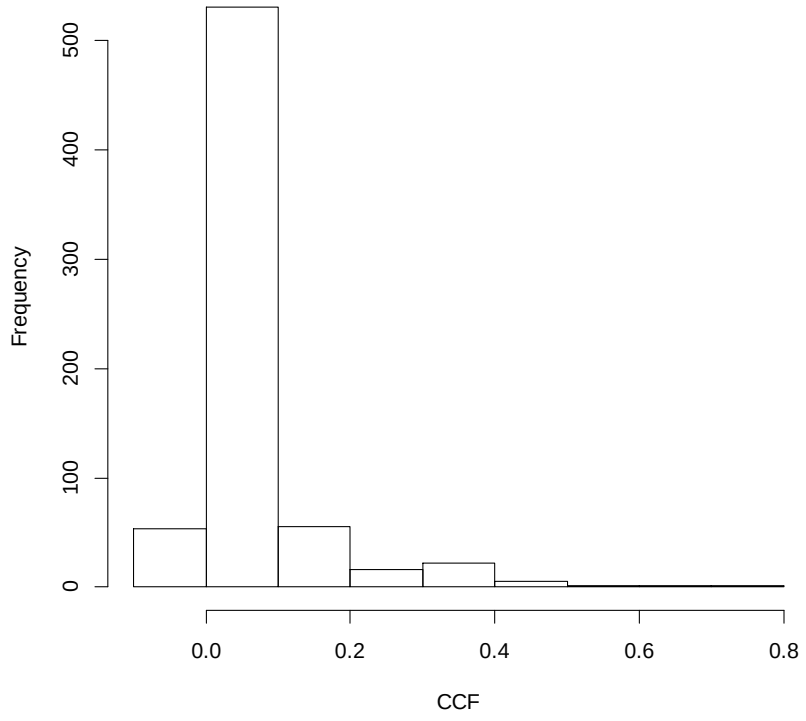


Figure 5.11 – Histogram of CCF values at temporal lag zero and spatial lags 1 and 2 combined

It can be seen that the majority of CCF values are positive but very weak (below the 0.25 threshold for weak correlation). The proportion of links that exhibit weak to strong positive CC with their neighbours is low. There are no links that exhibit marked CC.

5.3.4.5 Spatio-temporal pattern of cross-correlation

Figure 5.12 shows the relationship between cross-correlation, distance and temporal lag. There is a pattern of decreasing cross-correlation with increasing distance and temporal lag, which is to be expected. The peak cross-correlation is observed at temporal lag zero, indicating that the strongest correlation is contemporaneous. However, there are significant

cross-correlations at higher temporal orders, indicating that the information from the recent past is likely to be informative from a forecasting perspective. It can be seen that the distance decay is very sharp, falling towards zero at a range of approximately 2-3km.

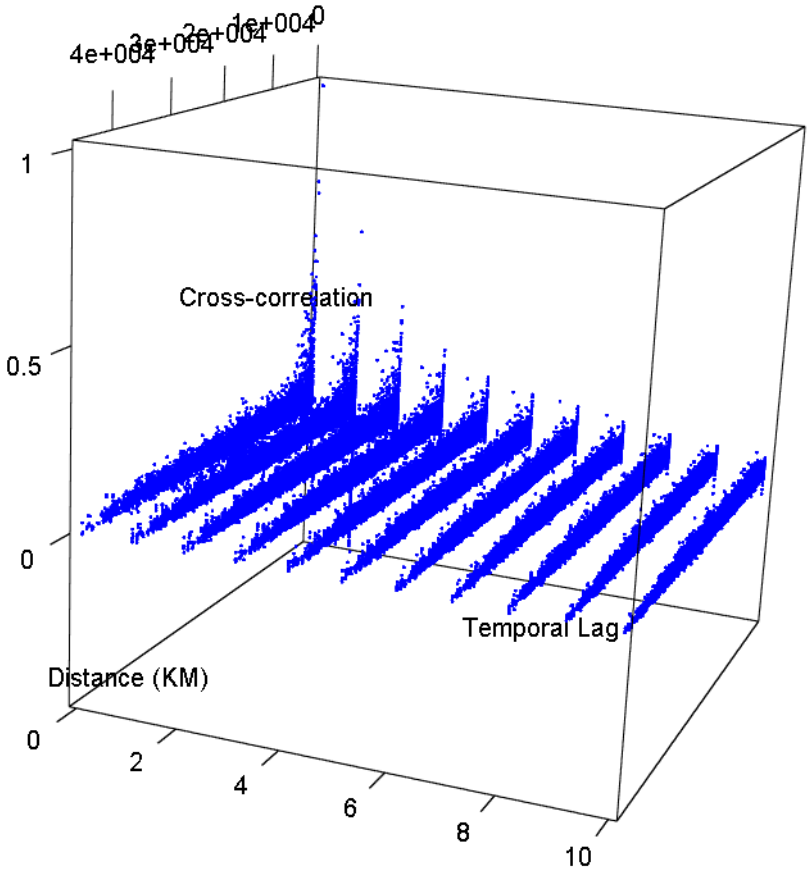


Figure 5.12 – 3D scatter plot of the relationship between correlation (z-axis), distance (x-axis) and temporal lag (y-axis).

5.3.5 Modelling implications of space-time autocorrelation structure

The temporal autcorrelation in UTTs recorded on the LCAP network is strong and positive. It is also seasonal, with daily and weekly components. This result motivates the use of temporal autoregressive type models with seasonal components.

The spatio-temporal CC between locations is generally positive at short spatial separation distances, with rapid distance decay. Although the CC is significant according to standard statistical tests, it is very weak compared with levels reported in some other studies, where

coefficients can be moderate to high (Yue and Yeh, 2008). The heterogeneity in the CC between locations clearly indicates that a spatially local model structure is needed, and global models such as STARIMA are unlikely to be sufficient. This has been demonstrated in Cheng et al (2013) on the same network. Furthermore, there is little evidence of peak cross-correlations occurring at nonzero temporal lags on a consistent basis, indicating that consistent directed relationships between locations cannot be elicited from the data. This makes the application of Bayesian network and state-space type models that are built on such relationships difficult.

Taking the results of the temporal and spatio-temporal CC analysis together, it can be concluded that temporal autocorrelation dominates over spatial autocorrelation in the data. Therefore, a purely temporal model structure that takes into account the time varying characteristics of the data is likely to be successful. However, the spatial neighbourhood may still provide useful information, even when CC is weak, and previous studies have been applied to networks where the level of correlation is weak (Kamarianakis and Prastacos, 2005; Chandra and Al-Deek, 2008). CC is a linear measure, and only represents the average relationship between two time series. It has been demonstrated in Cheng et al. (2011) on a subset of the LCAP network that the strength of spatio-temporal autocorrelation varies with time of day. It is possible for links to become more or less correlated with one another at different times and during different traffic conditions. Furthermore, the direction of influence can change. Although this phenomenon has not been systematically investigated in this thesis, a general recommendation can be made that a forecasting model should be capable of modelling the local dependencies in the data. This is the approach that is taken in models such as Min and Wynter (2011) and Kamarianakis et al. (2012).

In this case, because of the weak CC, sparse spatial structure and lack of flow data, it is difficult to impose a structure based on traffic flow theory as in Min and Wynter. Given the heterogeneity in the spatio-temporal relationships between locations, it is likely that a nonlinear machine learning approach will yield better results than a linear approach. ANNs have been shown in Chapter 3 to perform strongly in modeling the nonlinear characteristics of traffic data. However, the lack of interpretability of their internal structure and problems with local minima are limitations. KMs avoid these problems while maintaining the ability to model nonlinear structures, and were shown in table 3.4 to exhibit superior performance over ANNs in a number of studies. Therefore, the decision is made to develop a kernel based model with a local structure in the following Chapters.

5.4 Requirements for space-time forecasting models of network data

In this section, some requirements are formulated for a space-time forecasting model of network data, based on the outcomes of the literature review in Chapter 3, and the exploratory data analysis carried out in Chapters 4 and 5. These requirements inform the development of the models in Chapters 6 and 7, and are divided into two categories: 1) Generic requirements that apply to all space-time models of network data, and 2) Specific requirements of a TT forecasting model of the LCAP network.

Generic Requirements

1 **Accuracy and reliability:** The model should be able to forecast with an *acceptable degree of accuracy* on a *consistent basis*. The term *acceptable* is ambiguous. In this context it is evaluated based on two criteria: 1) performance compared with a naïve model, and; 2) performance compared with commonly used benchmark models.

2 **Dealing with missing and corrupt data in real time:** There are a range of methods for imputing missing data after the event, but a forecasting model should be *robust to the presence of missing data* in real time so as to maintain accuracy and reliability when sensors fail.

3 **Forecasting sufficiently far into the future so as to be fit for purpose:** The model should be capable of *accurate and reliable forecasts multiple steps into the future*, as required by the application.

4 **Maintaining computational efficiency for implementation on large networks in real time:** The model should be applicable to large networks in real time.

5 **Dealing with the temporally and spatially varying characteristics of network data, and adapting to changes in the distribution of the data:** A network model should be able to model the spatially and temporally varying characteristics of data, both in the short term and the long term, in order to ensure accuracy and reliability are maintained.

6 **Being as simple to understand and implement for practitioners as possible:** The model should be understandable, and intuitive to use if possible.

Application Specific Requirements

7 **Dealing with data that are MNAR:** The model should be able to deal with extended periods of missing data, including during peak times when accurate forecasts are most important.

8 **Forecasting for ATIS:** The model should be able to forecast travel times up to 1 hour ahead with reasonable accuracy, in order to be applicable to ATIS and other systems.

9 **Applicable to city wide forecasting:** The model should be capable of operating in real time on the whole LCAP network (~100+ links).

10 **Dealing with sparsely located sensor network and weak spatio-temporal CC:** The model should accommodate the spatial structure of the LCAP network, and model the heterogeneity in the spatio-temporal relationship between locations. Furthermore, it should be capable of producing reasonable forecasts in the presence of weak CC.

5.5 Chapter Summary

In this Chapter, the representation of the LCAP network as a graph has been described. The adjacency structure of the LCAP and the underlying road network are combined to produce a more accurate representation of the spatial relationship between sensor locations. This structure can be used in forecasting models. A space-time autocorrelation and CC analysis has been carried out to analyse the spatio-temporal relationship between links on the network. Based on the autocorrelation analysis, and reference to the literature discussed in Chapter 3, some model requirements have been defined for a space-time model of network data, and more specifically for a space-time model of the LCAP network. In the following Chapter, a temporal model is described that accounts for the time varying characteristics of network data. This model is extended to accommodate the spatio-temporal structure in the data in Chapter 7.

Chapter 6 - Local Online Kernel Ridge Regression

6.1 Chapter Overview

In Chapter 5, the network structure of the LCAP has been defined and an autocorrelation analysis has been carried out to assess the strength of temporal and spatio-temporal autocorrelation in travel times on the LCAP network. The network is characterised by strong, seasonal temporal autocorrelation and heterogeneous spatio-temporal autocorrelation that is generally weak. This motivates the use of a spatially and temporally local model structure. The model development is divided between two chapters. In this chapter, a temporally local model is developed for univariate forecasting of seasonal time series. The spatio-temporal extension of the model to the network is described in Chapter 7.

6.2 Introduction

Traditional time series forecasting models such as ARIMA require series to be stationary. That is, they must have constant mean and variance. Typically, the value of a series at the current time *t* is modelled as a linear or nonlinear combination of *m* previous observations at the same location in an autoregressive model. This structure is usually assumed not to change throughout time, i.e. if the current value is related to *m* observations at 9am, then it is related to *m* observations at 9pm in the same way. However, it has been discussed in the preceding chapters that, in many cases, the current value of a series is related nonlinearly to its previous values, and a globally fixed parametric structure is insufficient.

Kernel methods are a set of tools for nonlinear machine learning problems, including classification and regression (Shawe-Taylor and Cristianini, 2004). They make use of a kernel induced feature space constructed from a database of historical data patterns to store the relevant information about a particular problem. This feature space must contain sufficient richness of patterns in order to produce accurate forecasts, while not being so large as to sacrifice computational efficiency. In long time series, it is not feasible to use the entire historical dataset to forecast at each point in time because this would involve the construction of very large kernels. Therefore, typically one seeks to select a subset of the data that will be most useful. The most common approach is to use a small subset of the most recent observations, which makes the unrealistic assumption that a sufficiently diverse range of traffic states have been observed recently. For example, Wu et al. (2004) use just 5

weeks of data to train their SVR model, while Hong (2010, 2011a, 2012) and Hong et al. (2011) use just one month of data. While the ability to produce strong performance on smaller training data sets is one of the advantages of SVR (Vanajakshi and Rilett, 2004), the use of such a short period of training data means that there is unlikely to be sufficient richness of unusual traffic patterns.

Many time series exhibit very strong cyclic patterns. For instance, in the case of traffic data, there are usually two peaks, one in the morning and one in the evening, with an intervening period of lower traffic. This has the implication that data pertaining to the period between peaks will be largely ineffective in forecasting the peak periods, and vice versa. This motivates the development of models that produce forecasts based on similar traffic states, which is the approach that has been taken in, for example, Min and Wynter (2011) and Kamarianakis et al. (2012). Another approach that can be taken is to forecast the value at time t based on the similarity of the m previous points to the subset of patterns that have been observed at the same time on previous days. This approach makes the implicit assumption of similar traffic states at similar times of day and has the benefit of restricting the size of the pattern set to be evaluated against. However, it is likely to be overly restrictive.

Approaching the problem from the context of traffic forecasting, if one were to travel on the same section of road at approximately the same time each day, one may observe that it tends to become congested at approximately the same time each day. However, there would be significant variation around this trend. For instance, on some days the road section may become congested slightly earlier or later than usual; or the congestion may be slightly more or less severe. From the perspective of pattern analysis, this means that there is a temporal window around each point within which past information is informative, and this temporal window extends in both directions. The size of this window may be link dependent. On links where the duration of the congested period is shorter, it may be narrower. Furthermore, it may be time dependent. If a link has a longer duration AM peak than PM peak, for instance, then the window size may be larger in the AM peak than the PM peak. This means that a **temporal neighbourhood** can be defined for each time point. The temporal neighbourhood is the neighbourhood of temporal points that are relevant to the current point to be forecast.

This chapter outlines the specification of a local online kernel ridge regression (LOKRR) model that can capture the local temporal neighbourhood of seasonal time series. The

model is based on a nonlinear regression technique, namely kernel ridge regression (KRR). KRR uses the kernel trick to map the input data into a high dimensional feature space. In this space, a form of regularised linear regression called ridge regression (RR) is performed on the data. When the data is projected back to the input space, the resulting pattern function is nonlinear. The proposed model is online, which means that it is trained and updated in a sequential manner. Furthermore, it is temporally local, meaning that separate kernel parameters are trained for different points in time.

6.3 Kernels and kernel methods

The term *kernel method* is an umbrella term for a broad set of techniques that share a common characteristic. They comprise two components: 1) a function that maps the input data into a high (possibly infinite) dimensional space, known as a feature space, and; 2) a learning algorithm capable of discovering linear patterns in that space (Shawe-Taylor and Cristianini, 2004). Mapping to the feature space is accomplished efficiently using a *kernel function*, hence the term *kernel method*. Because linear relations are sought in the feature space, a broad range of theoretically well founded and efficient linear algorithms can be used. To date, many linear algorithms have been *kernelised* including ridge regression (Hoerl and Kennard, 1970; Saunders et al., 1998), the generalised portrait (Vapnik and Lerner, 1963; Boser et al., 1992), principal components analysis (Schölkopf et al., 1997) and canonical correlation analysis (Hotelling, 1936; Hardoon et al., 2004) amongst many others. Kernel methods are modular in nature, meaning that any kernel algorithm can be applied to a particular kernel, and vice versa (Shawe-Taylor and Cristianini, 2004). This gives them great flexibility as a tool for solving a wide range of practical problems.

In this section, the fundamentals of kernel methods are described. Firstly, in section 6.3.1, kernels are defined, and Gram and kernel matrices are described. Following this, in section 6.3.2, some of the popular types of kernel that are used in the literature are introduced. Finally, the concept of regularization is introduced, which is instrumental in ensuring the generalisation performance of kernel methods. For brevity, only the material that is necessary for the explanation of the methodology is included. For a more comprehensive introduction to kernel and kernel methods, the reader is directed to the texts of Schölkopf et al. (1998), Cristianini and Shawe-Taylor (2000), Schölkopf and Smola (2002) and Shawe-Taylor and Cristianini (2004). The introductory chapters of the latter text provide a well-structured introduction to kernel methods, using OLS as a starting point. For a detailed

treatment of regularisation in the context of kernel methods, readers may consult Schölkopf and Smola (2002).

6.3.1 Kernels

A kernel function is a function K such that for all $\mathbf{x}, \mathbf{z} \in X$:

$$K(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle \tag{6.1}$$

Where $\boldsymbol{\phi}$ is a mapping from input space X to a feature space F, and $\langle \cdot, \cdot \rangle$ denotes an inner (or dot) product. Kernels are necessarily symmetric such that:

$$K(\mathbf{x}, \mathbf{z}) = K(\mathbf{z}, \mathbf{x}) \tag{6.2}$$

They also satisfy the Cauchy Schwartz inequality:

$$K^2(\mathbf{x}, \mathbf{z}) \leq K(\mathbf{x}, \mathbf{x}) K(\mathbf{z}, \mathbf{z}) \tag{6.3}$$

To be a valid kernel, a symmetric function $K(\mathbf{x}, \mathbf{z})$ must be positive (semi) definite, which means that, for any set of examples $\mathbf{x}_1, \ldots, \mathbf{x}_l$ and any set of real numbers $\lambda_1, \ldots \lambda_l$, K must satisfy:

$$\sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j K(\mathbf{x_i}, \mathbf{x_j}) \geq 0 \tag{6.4}$$

Such matrices have non-negative eigenvalues. The term kernel is pervasive in the machine learning literature, but they are essentially covariances, which are well understood in the statistics literature (Genton, 2002). Kernels can be combined to make more complicated functions. The following are all admissible kernels (Cristianini and Shawe-Taylor, 2000):

1. $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) + K_2(\mathbf{x}, \mathbf{z})$
2. $K(\mathbf{x}, \mathbf{z}) = aK_1(\mathbf{x}, \mathbf{z})$
3. $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) K_2(\mathbf{x}, \mathbf{z})$
4. $K(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) f(\mathbf{z})$

5. $K(\mathbf{x}, \mathbf{z}) = K_3(\phi(\mathbf{x}), \phi(\mathbf{z}))$
6. $K(\mathbf{x}, \mathbf{z}) = \mathbf{x}'\mathbf{B}\mathbf{z}$

6.3.2 The Gram matrix and the kernel matrix

Given a set of vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, the Gram matrix is an $n * n$ matrix $\mathbf{G}$ with entries $\mathbf{G}_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$:

$$\mathbf{G} = \begin{pmatrix} \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_1, \mathbf{x}_n \rangle \\ \langle \mathbf{x}_2, \mathbf{x}_1 \rangle & \langle \mathbf{x}_2, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_2, \mathbf{x}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{x}_n, \mathbf{x}_1 \rangle & \langle \mathbf{x}_n, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_n, \mathbf{x}_n \rangle \end{pmatrix} \tag{6.5}$$

If the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ is replaced with a kernel product, then $\mathbf{G}$ is referred to as a kernel matrix $\mathbf{K}$ and has entries $\mathbf{K}_{i,j} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$:

$$\mathbf{K} = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \cdots & K(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & K(\mathbf{x}_n, \mathbf{x}_2) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \tag{6.6}$$

6.3.3 Types of kernel

Various types of kernel have been described in the literature and applied to a wide range of problem domains. The linear kernel is the most basic form of kernel and is simply the inner product between the input vectors:

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle \tag{6.7}$$

This kernel can be used if the relationship between the inputs **X** and the outputs **y** is linear. Using a linear kernel is equivalent to using the linear version of the learning algorithm. For example, in the context of ridge regression (RR), the linear kernel produces identical results to standard linear RR. However, the linear kernel can be useful from a computational point of view. In cases where the number of variables p is larger than the number of observations n, it is computationally more efficient to use the kernelised version of an algorithm than its linear version.

Another widely used kernel is the homogeneous polynomial kernel of degree d:

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^d \tag{6.8}$$

And its inhomogeneous version:

$$K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + c)^d \tag{6.9}$$

Where c is a non-zero constant.

The kernel that is most widely in the literature and has shown best performance in a wide range of application fields is the Gaussian radial basis function (RBF) kernel:

$$K(\mathbf{x}, \mathbf{z}) = exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \tag{6.10}$$

Where σ is the bandwidth of the kernel. The choice of an appropriate kernel function K is of great importance to the performance of kernel based algorithms. Although there is a wide range of kernels available, the RBF kernel has consistently been shown to be the best performing kernel in a wide range of applications and is used by convention.

6.3.4 Regularisation in kernel methods

Kernel methods share a problem with other nonlinear learning algorithms such as ANNs that the function space from which one can choose a solution is typically very large, or infinite. In order to choose a suitable function, one must have some prior knowledge of the function to be reconstructed. One assumption that can be made is the solution is smooth (Girosi et al., 1995). Therefore, one can consider a suitable function as one that fits the data with a small error whilst also being smooth. Regularisation is a term for a set of techniques that impose smoothness, or flatness, on the solution of function approximations.

Regularisation originated as a method for solving ill-posed problems, and has since become a method for preventing overfitting in machine learning models. It involves penalising a model based on its complexity, or number of parameters. Regularisation was first described in the context of solving an integral equation numerically by Tikhonov (1943). The same method is usually referred to as ridge regression in the statistics literature (Hoerl and Kennard, 1970), and has also been described as regularized least squares (Rifkin et al., 2003). In ridge regression, a constant $\lambda > 0$ is added to the diagonal of the matrix $\mathbf{X}'\mathbf{X}$ to solve the normal equations. In this context, Regularisation allows non-invertible (i.e. ill-conditioned or singular) matrices to be inverted, thus ensuring a solution for the parameters can be found. This is a form of L_2 norm regularisation. A more recent form of regularised linear regression is the least absolute shrinkage and selection operator (LASSO), which penalises the L_1 norm (Tibshirani, 1996). The practical difference between ridge regression and LASSO is that LASSO shrinks the parameters towards zero, and thus simultaneously carries out feature selection and regression. In the statistics literature, the Akaike Information Criterion (AIC) (Akaike, 1974) is a regularisation method commonly used for parameter selection in statistical models that controls the tradeoff between bias and variance, penalising the L_0 norm.

Kernel methods are suited to two types of regularisation: 1) a coefficient space constraint on the expansion coefficients of the weight vector in feature space, and; 2) a function space regularization that directly penalizes the weight vector in feature space (Schölkopf and Smola, 2002, p. 87). The well-known SVM is an example of the first type, which employs a regularisation constant, C, with a hinge loss function (Burges, 1998). The regression version, ε-SVR, uses the same constant, with an ε-insensitive loss function. In the latter case, C determines the trade-off between the flatness (smoothness) of the function and the amount up to which deviations of larger than ε are tolerated (Smola and Schölkopf, 2004). KRR and Gaussian Processes Regression (GPR) are examples of the second type, in which a regularisation constant is used to penalize large weights.

6.4 Local Online Kernel Ridge Regression

In this section, the method of local online kernel ridge regression (LOKRR) is described. In section 6.4.1, linear ridge regression is introduced, which forms the basis for the nonlinear kernelised version, which is described in section 6.4.2. In section 6.4.3, the online version of KRR, OKRR, is described, which is based on the sliding window kernel recursive least squares

method (KRLS) of van Vaerenbergh et al. (2006). Finally, the temporally local version is described in section 6.4.4.

6.4.1 Ridge Regression

Beginning with the well-known case of multiple linear regression, one seeks to solve a system of equations of the following form:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon} \tag{6.11}$$

Where $\mathbf{X}$ is $(n * p)$ and of rank p, n is the number of observations, p is the number of variables, $\mathbf{w}$ is $(p * 1)$ and unknown, $E[\boldsymbol{\varepsilon}] = 0$, and $E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'] = \sigma^2\mathbf{I}_n$. The solution of this system for $\mathbf{w}$ is:

$$\mathbf{w} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{6.12}$$

This solution is viable when $\mathbf{X}'\mathbf{X}$ is nearly a unit matrix. However, if this is not the case then this solution is sensitive to the number of errors in the data. To overcome this problem, a regularisation constant λ can be introduced:

$$\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{y};\ \lambda \geq 0 \tag{6.13}$$

λ is called the ridge parameter and the resulting algorithm is known as ridge regression. The matrix $(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_p)$ is always invertible if $\lambda > 0$, allowing the solution of ill-posed regression problems. If $\lambda = 0$, equation 6.13 is equivalent to equation 6.12.

6.4.2 Kernel Ridge Regression

Equation 6.13 can be rearranged in terms of $\mathbf{w}$ as follows (Shawe-Taylor and Cristianini, 2004):

$$\mathbf{w} = \lambda^{-1}\mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{X}'\boldsymbol{\alpha} \tag{6.14}$$

$\mathbf{w}$ can be written as a linear combination of the training data points, $\mathbf{w} = \sum_{i=1}^{n} a_i \mathbf{x}_i$, with $\boldsymbol{\alpha} = \lambda^{-1}(\mathbf{y} - \mathbf{X}\mathbf{w})$. Therefore, $\boldsymbol{\alpha}$ can be computed as:

$$\boldsymbol{\alpha} = \lambda^{-1}(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\Rightarrow \lambda\boldsymbol{\alpha} = (\mathbf{y} - \mathbf{X}\mathbf{X}'\boldsymbol{\alpha})$$

$$\Rightarrow (\mathbf{X}\mathbf{X}' + \lambda\mathbf{I_n})\boldsymbol{\alpha} = \mathbf{y}$$

$$\Rightarrow \boldsymbol{\alpha} = (\mathbf{G} + \lambda\mathbf{I_n})^{-1}\mathbf{y}$$

(6.15)

Where $\mathbf{G} = \mathbf{X}\mathbf{X}'$, and is the Gram matrix introduced in section 6.3.2. It can be seen from equation 6.15 that the solution is now written in terms of the data points and the weight vector $\mathbf{w}$ need not be solved explicitly. This formulation is computationally more efficient than equation 6.13 when $p > n$, which is often the case in machine learning applications. However, the main benefit of expressing the algorithm in this way is that, given a valid kernel function K, $\mathbf{G}$ can be replaced with a kernel matrix $\mathbf{K}$ as follows:

$$\alpha = (\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{y}$$

(6.16)

This algorithm is known as kernel ridge regression (KRR). A new data point can be computed as:

$$g(\mathbf{x}) = \mathbf{y}'(\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{k}$$

(6.17)

Where $\mathbf{K}$ is a kernel matrix of inner products between training vectors and $\mathbf{k} = k(\mathbf{x}_i, \mathbf{x})$ is a vector of inner products between the test vector $\mathbf{x}_i$ and the training vectors $\mathbf{x}$.. KRR is called a regularization network in the ANN literature (Poggio and Girosi, 1990). The method is also strongly related to Kriging (Krige, 1951), which is termed GPR in the machine learning community (Rasmussen and Williams, 2006). Kriging has been widely applied in the spatial sciences. The explicit difference between Kriging and KRR is that Kriging uses a covariance function to model the dependence between observations recorded at spatial (and temporal) coordinates. The coordinates themselves are the independent variables, and the covariance function is usually estimated from the data apriori, often in idealized form (Kyriakidis and Journel, 1999; Gneiting, 2002). In KRR, the kernel function need not be a covariance

function, and can be any valid kernel function. It is not necessary to use the spatial (and temporal) coordinates as the independent variables, and the measure of similarity (kernel) is computed between the input vectors themselves. This allows the inclusion of exogenous variables.

6.4.3 Online Kernel Ridge Regression

There are two distinct ways of training a kernel algorithm. The first, known as batch training, involves processing all of the training examples at once and finding a suitable model. This method relies on the training data explaining enough of the variance in the data in order to generalise well to new data. The second, known as online training, involves processing the data one example at a time, making a forecast, and then updating the model based on the predictive error (Shawe-Taylor and Cristianini, 2004). Online learning is useful in situations where data arrive one sample at a time in a continuous stream, which is the case in time series applications. Batch training using time series data assumes two things: 1) enough data are available at the time of training to construct a representative model, and; 2) the trained model will retain its generalisation ability over time. However, it is well know that time series exhibit temporal dependence and are often nonstationary. Data collected in the recent past is more informative than data collected in the distant past, which is the theoretical basis for building autoregressive models. It is desirable, therefore, for a pattern based time series model to retain new data patterns as they arrive, and forget old data patterns as they lose their relevance. This motivates the use of an online model structure.

The model described here is based on the sliding window kernel recursive least squares (KRLS) algorithm of van Vaerenbergh et al. (2006), which is equivalent to an online KRR (OKRR) algorithm. Given a stream of training data points $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots\}$, the training data at time t is constructed as $\mathbf{y}_t = [y_t, y_{t-1}, \dots, y_{t-N+1}]'$ and $\mathbf{X}_t = [\mathbf{x}_t, \mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-N+1}]'$, where y_t and $\mathbf{x}_t$ are the values of the dependent and independent variables at time t, respectively, N is the size of the window and $t = 1,2,..N$. A regularised kernel matrix $\mathbf{K}_t^{\lambda} = (\mathbf{K}_t + \lambda \mathbf{I}_t)$ is constructed from $\mathbf{X}_t$. At the following time step, the data window slides along by one point to time $t + 1$ and the most recent observation is added to the training data and the oldest training point is removed. A new matrix $\mathbf{K}_{t+1}^{\lambda}$is constructed. Thus, at each time point the kernel matrix is constructed from only the previous N training examples.

The main computational burden of KRR lies in calculating the inverse of the regularised kernel matrix in equation 6.17. OKRR requires the inverse to be recalculated at each time

step, which is infeasible in an online setting. Fortunately, methods exist for updating $\mathbf{K}_t^{\lambda^{-1}}$ without computing the inverse from scratch. To remove a row and column, a kernel matrix $\mathbf{K}$ and its inverse $\mathbf{K^{-1}}$ can be partitioned as follows (van Vaerenbergh et al. 2006, p.8):

$$\mathbf{K} = \begin{bmatrix} a & \mathbf{b}' \\ \mathbf{b} & \mathbf{D} \end{bmatrix}, \mathbf{K}^{-1} = \begin{bmatrix} e & \mathbf{f}' \\ \mathbf{f} & \mathbf{G} \end{bmatrix}$$

(6.18)

The inverse of the submatrix $\mathbf{D}$ is required, which can be calculated from the submatrices of $\mathbf{K}^{-1}$ as follows:

$$\mathbf{D}^{-1} = \mathbf{G} - \mathbf{f}\mathbf{f}'/e$$

(6.19)

Therefore, the updated inverse is calculated from the known elements of $\mathbf{K}^{-1}$. To add a row and column to the inverse, $\mathbf{K}^{-1}$ and $\mathbf{K}$ are partitioned as follows:

$$\mathbf{K} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}' & d \end{bmatrix}, \mathbf{K}^{-1} = \begin{bmatrix} \mathbf{E} & \mathbf{f} \\ \mathbf{f}' & g \end{bmatrix}$$

(6.20)

Then one can take advantage of the following formula:

$$\mathbf{K}^{-1} = \begin{bmatrix} \mathbf{A}^{-1}(\mathbf{I} + \mathbf{b}\mathbf{b}'\mathbf{A}^{-1H}g) & -\mathbf{A}^{-1}\mathbf{b}g \\ -(\mathbf{A}^{-1}\mathbf{b})'g & g \end{bmatrix}$$

(6.21)

Where:

$$g = (d - \mathbf{b}'\mathbf{A}^{-1}\mathbf{b})^{-1}$$

(6.22)

Here, the superscript H of $\mathbf{A}^H$ denotes the conjugate transpose of $\mathbf{A}$. $\mathbf{A}^{-1}$ is the inverse kernel matrix obtained at the previous time step, demonstrating that the inverse kernel matrix does not need to be recalculated from scratch. These steps reduce the computational complexity of inverting the matrix from $O(N^3)$ to $O(N^2)$, where N is the number of data samples in the kernel.

6.4.4 Local Online Kernel Ridge Regression

OKRR has the advantage over standard KRR that it can incorporate new information in the model structure without a significant increase in computation time. However, it has two limitations that may affect its application to certain processes. Firstly, a single kernel is constructed using only the independent variable vectors of the N time points immediately preceding time t. This is reasonable in cases where time series exhibit sudden regime changes such as the Wiener system reported in van Vaerenbergh et al. (2006). However, in seasonal data, this is likely to be insufficient since the most informative data comes not just from the immediately preceding time points, but from the same times on previous days, weeks or months. The second drawback is that it uses a single set of parameters (ridge parameter λ and kernel parameter(s)) to describe the behaviour of the system across all times. This limits the ability of the model to account for heteroskedasticity. Many real world network processes exhibit heteroskedasticity. For example, the variance in traffic data is higher in the peak periods. To address these two limitations, a local version of OKRR is proposed, namely Local (L)OKRR, in which a separate kernel is defined for each time of day, each with its own set of parameters.

6.4.4.1 Motivation of the model

To motivate the model in the current context, the example of forecasting travel times on a single road link is used. Assume a training dataset of travel time patterns obtained at 5 minute intervals over a number of days. Each day, (60/5)*24 = 288 observations are made. Now assume that because of computational constraints the maximum kernel size is set to 10,000*10,000. Using the single kernel approach of van van Vaerenbergh et al. (2006), this limit would entail a maximum of 10,000/288 = 34.7 days of data could be included in the kernel. This approach has two clear drawbacks. Firstly, 34.7 days is a relatively short time in this context, and it is unlikely that sufficient variation in traffic patterns will have been observed over this period in order to produce reliable forecasts, particularly under abnormal conditions. Secondly, because of the strong cyclic pattern present in traffic data, historical traffic patterns recorded at temporal lags that are either close to zero or close to divisible by 288 are likely to be much more informative than those at other lags.

Consider as an alternative, the extreme example where the next travel time is forecast as a function of only those travel times that were observed at the same time of day on previous days. With the same maximum kernel size, one could build a kernel containing the travel

time patterns obtained on the previous 10,000 days. Of course, in practice, one would not have access to 10,000 days (27.4 years) of historical data, and the nonstationarity in long term travel patterns would render data collected beyond a certain time threshold irrelevant. However, this example serves to highlight the fact that if knowledge of seasonality is directly incorporated into the model structure, then there is potential for having access to a much richer source of information about the time point to be forecast without increasing the kernel size. In fact, it is possible to include much more informative data in a far smaller kernel. Using the same example, one could capture the behaviour of the link at a single time point over the course of a year in a kernel of size 365*365. To this end, local temporal kernels are proposed as a methodology for incorporating seasonality into kernel based regression models. Although KRR is used as the algorithm here, the general model structure could be easily adapted to other kernel methods, such as SVR and GPR.

6.4.4.2 Local temporal kernels

First, the case outlined in the previous paragraph of using just the data points collected at the same time on previous days is described. Consider, again, a stream of training data points $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots\}$. The goal is to forecast $\mathbf{y}_t$ using a subset of the observed patterns. Assume the dataset has a regular seasonal component with order S. The training dataset is constructed as $\mathbf{y}_{st} = [y_{1,t}, y_{2,t}, \dots, y_{N,t}]'$ and $\mathbf{X}_{st} = [\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \dots, \mathbf{x}_{N,t}]'$, where $s = 1,2, \dots, N$ is the index of the seasonal component, for example the day index, and $t = 1,2, \dots, S$ is the time of day index. This formulation is shown diagrammatically in figure 6.1.

	t_1	t_2	…	t_s
s_1	$(\mathbf{x}_{11}, y_{11})$	$(\mathbf{x}_{12}, y_{12})$	…	$(\mathbf{x}_{1S}, y_{1S})$
s_2	$(\mathbf{x}_{21}, y_{21})$	$(\mathbf{x}_{22}, y_{22})$	…	$(\mathbf{x}_{2S}, y_{2S})$
⋮	⋮	⋮	⋱	⋮
s_N	$(\mathbf{x}_{N1}, y_{N1})$	$(\mathbf{x}_{N2}, y_{N2})$	…	$(\mathbf{x}_{NS}, y_{NS})$

Figure 6.1 – Diagram of the training data construction

The rows of the table indicate the successive days in the training dataset. The columns represent the successive time intervals in a day. N is the total number of days in the training dataset. In the example outlined above of forecasting travel times, $S = 288$. Under this formulation, S kernels $\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_S$ are constructed from the columns of figure 6.1. For

example, to forecast the travel time at t_2, the first column of training examples in is used. Compared with a single kernel model, this would appear to be a large number of kernels, however, each of them can have a much smaller dimension than a single kernel model. Furthermore, a smaller kernel needs to be updated at each time step, which is computationally more efficient than updating a single large kernel. Equation 6.23 shows a local temporal kernel constructed in this way.

$$\mathbf{K}_{st} = \begin{pmatrix} k(\mathbf{x}_{1,t}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{1,t}, \mathbf{x}_{2,t}) & \cdots & k(\mathbf{x}_{1,t}, \mathbf{x}_{N,t}) \\ k(\mathbf{x}_{2,t}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{2,t}, \mathbf{x}_{2,t}) & \cdots & k(\mathbf{x}_{2,t}, \mathbf{x}_{N,t}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_{N,t}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{N,t}, \mathbf{x}_{2,t}) & \cdots & k(\mathbf{x}_{N,t}, \mathbf{x}_{N,t}) \end{pmatrix} \tag{6.23}$$

6.4.4.3 Local temporal windows

The advantage of the formulation outlined above is that it directly incorporates seasonality into the model structure. However, it has the drawback that it may exclude a significant amount of data that may be of interest. Returning to the example of travel time forecasting on a road link, if one were to commute to work on the same road at approximately the same time each day, one may observe that the road tends to become congested at approximately the same time each day, and may be able to make statements such as “if I leave after 9am there is always too much traffic”, or “if I set off before 8am my journey is usually pretty quick”. However, there is usually significant variation around such trends. For instance, on some days a link may become congested earlier or later than usual; or the congestion may be slightly more or less severe, and one might find oneself making a statement such as “It’s especially busy today, something must have happened”, or “wow, it’s really quiet, it’s usually really busy by now”. From the perspective of pattern analysis, this means that there is a temporal window extending in both directions around each point within which past information is informative. This captures the seasonal autocorrelation and heteroskedasticity present in the data.

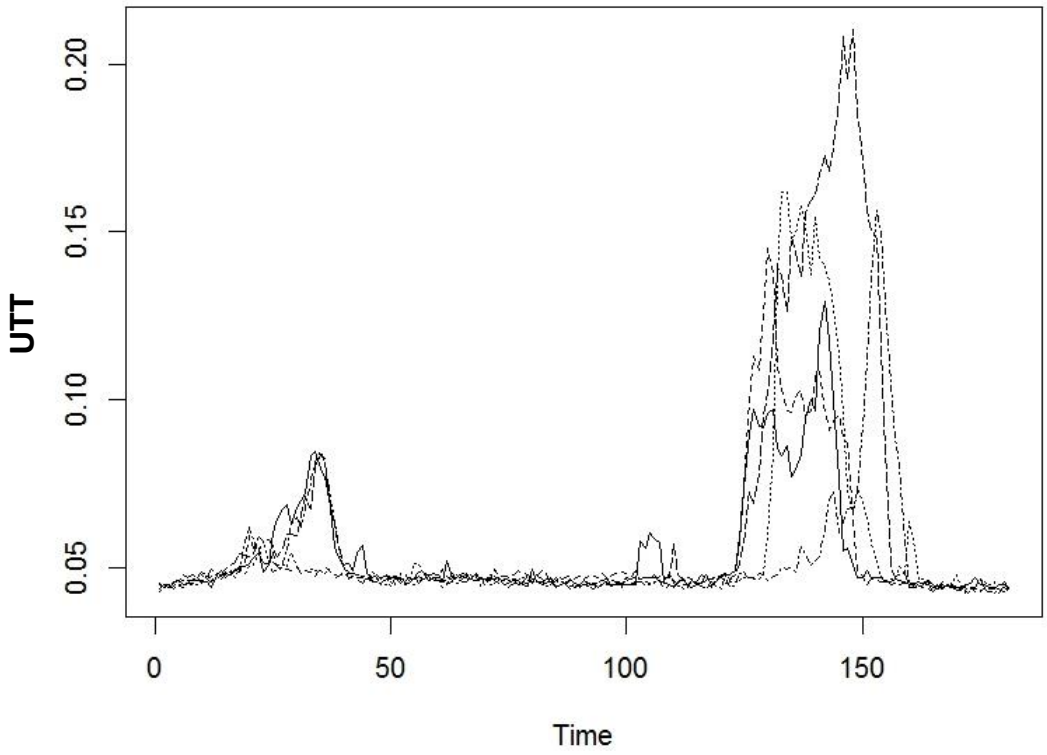


Figure 6.2 – Illustration of temporal window: Each line in the plot is the UTT profile recorded on a single link (link 1815) on a different day (5 days total).

Figure 6.2 shows an illustration of this phenomenon. There are two recurrent peaks on this link, a small one in the morning and a larger one in the evening. Although they occur at roughly the same time each day, the time at which they begin and end, as well as their magnitude, differs within a time window. Within this window, the observed historical patterns are likely to be informative and should be included in the kernel. To account for this, a local temporal window is formed around each time t, which is denoted as w and the training data is constructed as:

$$\mathbf{y}_{stw} = \left[\{y_{1,t-w}, \cdots, y_{1,t-1}, y_{1,t}, y_{1,t+1}, \cdots, y_{1,t+w}\}, \cdots, \{y_{n,t-w}, \cdots, y_{n,t-1}, y_{n,t}, y_{n,t+1}, \cdots, y_{n,t+w}\}\right]' \tag{6.24}$$

$$\mathbf{x}_{stw} = \left[\{\mathbf{x}_{1,t-w}, \cdots, \mathbf{x}_{1,t-1}, \mathbf{x}_{1,t}, \mathbf{x}_{1,t+1}, \cdots, \mathbf{x}_{1,t+w}\}, \cdots, \{\mathbf{x}_{n,t-w}, \cdots, \mathbf{x}_{n,t-1}, \mathbf{x}_{n,t}, \mathbf{x}_{n,t+1}, \cdots, \mathbf{x}_{n,t+w}\}\right]' \tag{6.25}$$

Based on this, kernels $K_1, K_2, \dots, K_s$ are defined according to:

$$K_t = \begin{pmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,n} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n,1} & K_{n,2} & \cdots & K_{n,n} \end{pmatrix}$$

(6.26)

Where:

$$K_{1,1} = \begin{pmatrix} k(\mathbf{x}_{1,t-w}, \mathbf{x}_{1,t-w}) & \cdots & k(\mathbf{x}_{1,t-w}, \mathbf{x}_{1,t-1}) & k(\mathbf{x}_{1,t-w}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{1,t-w}, \mathbf{x}_{1,t+1}) & \cdots & k(\mathbf{x}_{1,t-w}, \mathbf{x}_{1,t+w}) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_{1,t-1}, \mathbf{x}_{1,t-w}) & \cdots & k(\mathbf{x}_{1,t-1}, \mathbf{x}_{1,t-1}) & k(\mathbf{x}_{1,t-1}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{1,t-1}, \mathbf{x}_{1,t+1}) & \cdots & k(\mathbf{x}_{1,t-1}, \mathbf{x}_{1,t+w}) \\ k(\mathbf{x}_{1,t}, \mathbf{x}_{1,t-w}) & \cdots & k(\mathbf{x}_{1,t}, \mathbf{x}_{1,t-1}) & k(\mathbf{x}_{1,t}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{1,t}, \mathbf{x}_{1,t+1}) & \cdots & k(\mathbf{x}_{1,t}, \mathbf{x}_{1,t+w}) \\ k(\mathbf{x}_{1,t+1}, \mathbf{x}_{1,t-w}) & \cdots & k(\mathbf{x}_{1,t+1}, \mathbf{x}_{1,t-1}) & k(\mathbf{x}_{1,t+1}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{1,t+1}, \mathbf{x}_{1,t+1}) & \cdots & k(\mathbf{x}_{1,t+1}, \mathbf{x}_{1,t+w}) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_{1,t+w}, \mathbf{x}_{1,t-w}) & \cdots & k(\mathbf{x}_{1,t+w}, \mathbf{x}_{1,t-1}) & k(\mathbf{x}_{1,t+w}, \mathbf{x}_{1,t}) & k(\mathbf{x}_{1,t+w}, \mathbf{x}_{1,t+1}) & \cdots & k(\mathbf{x}_{1,t+w}, \mathbf{x}_{1,t+w}) \end{pmatrix}$$

(6.27)

Including a window of patterns around t increases the amount of local temporal information available to the model. However, it also increases the kernel size. The kernel defined in equation 6.23 is $n * n$, whereas the kernel defined in equation 6.27 is $(n * ((2 * w) + 1)) * (n * ((2 * w) + 1))$. As w increases, the dimension of $\mathbf{K}$ increases by $2n$. Therefore, it is preferable to keep w small. However, it should be noted that, in an online setting, this formulation is still efficient compared with using a kernel defined on all the data.

6.4.5 Parameter selection of LOKRR

The formulation outlined above allows for local tuning of the model parameters. Given a model with $s = 288$, 288 separate kernels are defined, each of which has its own set of parameters. The training of these parameters has no added computational cost over the training of a single set of parameters since the model still requires the inversion of a single kernel matrix at each time step. Therefore, each of the 288 models can be trained simultaneously. This enables the model to capture temporal heteroskedasticity by allowing the kernel bandwidth and the ridge parameter to vary by time of day. Parameter selection in kernel based models is an active research area, and various methods, mainly heuristic in nature, have been used to improve the parameter selection process. For example, in SVMs, genetic algorithms (Üstün et al., 2005; Wu et al., 2009; Cai et al., 2009; Li and Yang, 2008), chaotic immune algorithms (J. Wang et al., 2009), particle swarm optimisation (X. Li et al., 2010), immune particle swarm optimisation (Y. Wang et al., 2009) differential evolution

(Lahiri and Ghanta, 2008; Li and Cai, 2008) and ant colony optimisation (Zheng et al., 2008) have been used to select parameters, amongst others. Although this research has been instrumental in improving the speed of training of machine learning algorithms, in an applied setting it is desirable to have a parameter selection process that is interpretable and simple to implement, particularly when the model is to be applied in a large number of cases. In the following subsections, the selection method is described for each of the model parameters in the context of using a Gaussian RBF kernel, which are λ, σ and w. The parameter space is derived from the data, making parameter selection straightforward in any application.

6.4.5.1 Tuning λ

The regularisation constant λ is a free parameter that needs to be tuned. It plays a similar role to C in SVR. Usually, λ is determined by cross-validation over a space of values. It is sensible to try to limit this space of values in order to limit the computation time required to train models. Exterkate (2013) showed that appropriate values of λ can be found by relating λ to the signal to noise ratio φ. First the R^2 is obtained from an OLS fit of $\mathbf{y}$ on $\mathbf{X}$. If the OLS fit were the true model, then $\varphi_0 = R^2/(1-R^2)$. From this, λ_0 can be determined for a Gaussian kernel from the signal to noise ratio as $\varphi_0 = 1/\lambda_0$. Exterkate recommends the following grid be used to train λ: $\{\frac{1}{8}\lambda_0, \frac{1}{4}\lambda_0, \frac{1}{2}\lambda_0, \lambda_0, 2\lambda_0\}$. In the study, Monte Carlo simulation was used to compare the mean squared prediction error (MSPE) obtained from this grid with that obtained from the true values of λ, and it was found that estimating λ resulted in an increase in MSPE of only around 0.4%. It follows, therefore, that the added computational effort associated with training a larger grid of values for λ is not justified.

6.4.5.2 Tuning σ

The second free parameter in the KRR model is the kernel parameter. In the case of the Gaussian kernel, this is the kernel bandwidth σ, which determines the smoothness of the forecasting function. A large value of σ results in a smooth function while a small value results in a more complex function. There is a wealth of literature on the training of σ in Gaussian kernels, particularly in the context of SVMs, and some of the methods that have been used to date were listed in section 6.4.5. However, it is important to recognise that there is a small range of values within which σ can vary, which is based on the variance of the data in a given kernel. Caputo et al. (2002) showed that the optimal values of σ lie between the 0.1 and 0.9 quantiles of $\|\mathbf{x} - \mathbf{x}'\|^2$ and so can be estimated from the Euclidean

distance between the data points in the kernel. This is the scheme used for automatic σ estimation for SVM models in the R package Kernlab (Karatzoglou et al., 2006, 2004). Exterkate (2013) proposed a similar method for selecting σ in the context of KRR.

While it is likely that more sophisticated training schemes will yield slightly more accurate results, the size of the improvement is unlikely to justify the increased computational burden of training extra parameter combinations. Therefore, in this study, the approach of Caputo et al is used to estimate the median, 0.25 and 0.75 quantiles of each kernel, and these three values of σ are used to train each model. Not only does this reduce the size of the parameter space that needs to be searched, it also provides a principled way of training LOKRR models on other datasets.

6.4.5.3 Tuning the window size w

The window size w is an important parameter that decides how much local temporal information to include in the model. The larger w gets, the more information is included in the model pertaining to each day of the training data. However, as mentioned in section 6.4.4.3, an increase in w of 1 results in an increase in the dimension of the kernel of $2n$. Therefore, it has a significant effect on the computation time required to train and update models. Holding the overall size of the kernel static, an increase in w necessitates a $2w + 1$-fold decrease in n. Therefore, a smaller w is desirable in order to maintain computational efficiency. The value of w is determined in the model training process and depends on the application. However, it is recommended to restrict its maximum value. The values tested here are $w = 1,2,3$. Because of strong seasonal temporal autocorrelation in the data it is assumed that a maximum window size of 3 is sufficient. In other applications, the author recommends a strategy of holding the kernel size static, and reducing the training length as w increases in order to examine the tradeoff between w and training data length.

6.4.5.4 Data normalization

Data normalisation, also referred to as scaling, is important in kernel methods because it brings all of the independent variables into the same range. This ensures that no single variable dominates over the rest of the variables when the distance between the training vectors is calculated in the kernel. There are various types of normalisation that could be used, including standardization to produce z-scores, scaling by domain (i.e. [0,1]) and minmax normalisation (Juszczak et al., 2002). Each kernel is normalised separately. This

enables each kernel to capture the distribution of the data around time t, and brings the kernel parameters for each time point into the same range. This also enables the heteroskedasiticity in the data to be examined through the model parameters. In the context of traffic data, it would be expected that the variance in σ would be higher during the peak periods than in the inter peak periods.

6.5 Case Study: Forecasting travel times with LOKRR

In this section, the LOKRR model is tested in the context of forecasting travel times collected on the LCAP network. Firstly, in section 6.5.1, the data are described. Following this, in section 6.5.2, the model training procedure is outlined. The formulation of the time series model is described in section 6.5.3 and the forecasting scenarios are described in section 6.5.4. In section 6.5.5, the evaluation criteria are described. The benchmark models against which the LOKRR model is to be compared are described in section 6.5.6, before the results are given in section 6.6.

6.5.1 Data description

To test the temporal model, the ten LCAP links with the lowest patch rates (see Chapter 4) are selected. These links are shown in table 6.1, along with the average data frequency (number of vehicles per observation). The spatial location of the links in the network is shown in figure 6.3. As one would expect, high capture rates coincide with high frequency relative to the network average in these cases. Links with low levels of missing data are selected in order to minimise the effects of missing data on the results.

Table 6.1 – The test links and their patch rates and frequency

Link ID	**24**	**26**	**442**	**454**	**1815**	**1799**	**453**	**1798**	**2448**	**881**
% Patched	0.8	0.8	1.15	1.2	1.22	1.3	1.39	1.44	1.44	1.45
Avg. Frequency	20.95	15.49	14.09	11.74	68.82	31.69	13.74	28.53	9.37	18.96
Length (m)	366.7	907.5	899.4	710.1	4033.1	1101	892.6	4670.2	1216.5	3744.3

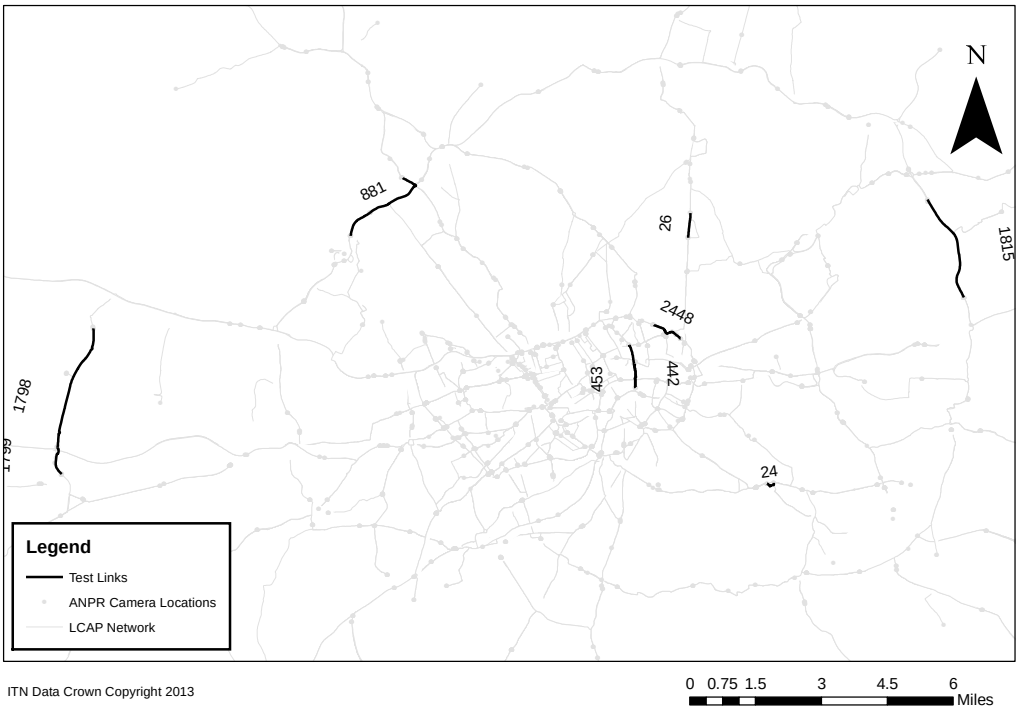


Figure 6.3 –Location of the test links on the LCAP network

Figure 6.4 shows the time series of each of the test links over the course of 10 weeks (weekends included). Evidence of the seasonal component in the data is present in the majority of links. However, there is significant variation from day to day. Higher unit journey times are observed in the AM and PM peak periods to varying degrees in all of the links, but there is clear heterogeneity in the magnitude (height) and the duration of the peaks from day to day. Between links there is also considerable variation in the observed traffic patterns. Links 26, 442, 2448 and 881 appear to have fairly stable traffic patterns. However, other links such as 1815 and 1799 are characterised by few very large peaks.

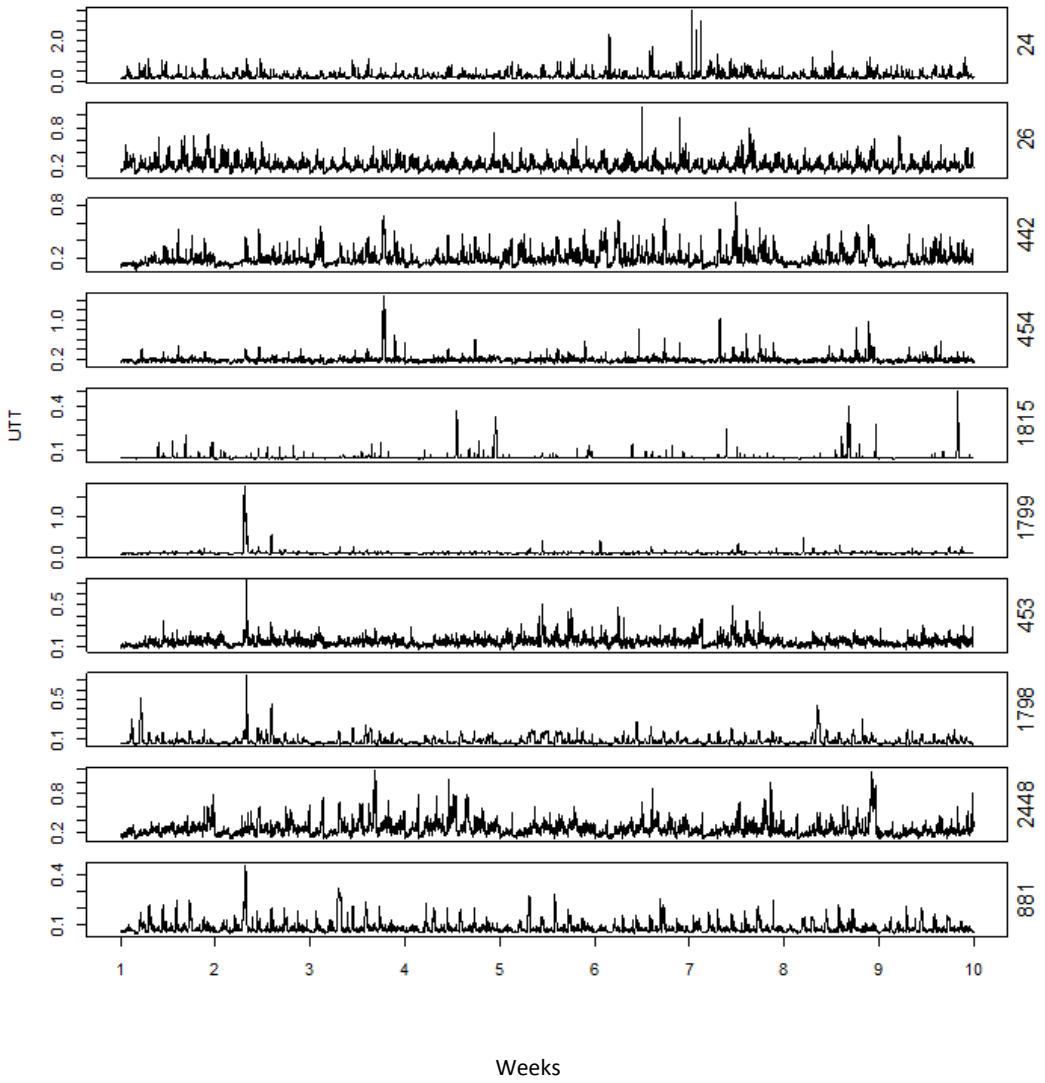


Figure 6.4 – Time series of each of the test links over the first 10 weeks of the training period

In total there are 254 days for which data are available, collected between January and October 2011. To test the models, the data are divided into three sets; a training set, a testing set and a validation set, which are 80 days (52%), 37 days (24%) and 37 days (24%) in length, respectively. The testing set runs from Tuesday 28th April to Wednesday 1st June 2011, and the validation set runs from Thursday 2nd June to Friday 8th July 2011. The sizes of the training and testing sets are fixed in this experiment for simple comparison of results.

6.5.2 Model Training

A sliding window validation approach is used to train the models. At the first time step, the training set is used to build a model to forecast the first day of the testing set. Following this,

the first day of the training set is removed and replaced with the first day of the testing set. A model is then built using the new training set to forecast the second day of the testing set, and so on and so forth. The selected model is the one that minimises the average training error across the days in the testing set. To measure the training error, the root mean squared error (RMSE) index is used. This is defined as follows:

$$RMSE = \sqrt{1/n \sum_{t=1}^{n} (y_t - \hat{y}_t)^2}$$

(6.28)

Where y_t and $\hat{y}_t$ are the observed and forecast values and n is number of observations.

6.5.3 Model Description

One of the most important decisions that must be made when creating a forecasting model is which variables to include. Here, an autoregressive model structure is used, which is the standard approach. In an autoregressive model, m previous observations of the series are used to forecast the next value, where m is the embedding dimension. Continuing the notation from section 6.4.3, assume a stream of data points $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots\}$ is to be arranged into a vector of dependent variables $\mathbf{y}_t = [y_t, y_{t-1}, \dots, y_{t-N+1}]'$ and a matrix of independent variables $\mathbf{X}_t = [\mathbf{x}_t, \mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-N+1}]'$. In an embedded time series, the independent variable vector at time t is constructed as follows[22]:

$$\mathbf{x}_t = [y_{t-1}, y_{t-2}, \dots, y_{t-m}]$$

(6.29)

In this formulation, the temporal ordering of the data is not made explicit beyond the definition of the window w. Therefore, all data patterns in the kernel are considered equally and a forecast is made based only on the similarity of the data pattern observed at time t with the patterns in the historical data. It has been demonstrated in previous studies that including a time varying average as a variable in the model can improve the predictive performance of pattern based models (Smith and Demetsky, 1996). The sample time varying average $\hat{\mu}_t$ can be calculated as:

[22] An intercept term is also added in the LOKRR model, but is omitted from the description both for clarity and because it is not required for all time series models.

$$\hat{\mu}_t = 1/n \sum_{i=1}^{n} y_{i,t}$$

(6.30)

Where n is the number of days in the training data. The mean is appended to the vector defined in equation 6.29 to give:

$$\mathbf{x}_t = [y_t, y_{t-1}, \dots, y_{t-m}, \hat{\mu}_t]$$

(6.31)

Including $\hat{\mu}_t$ in the model introduces the notion of temporal ordering into the feature space defined by the kernel. Essentially, this has the effect of assigning a prior assumption that those patterns observed at near times to the current pattern are likely to be more informative than those observed at different times. At longer forecasting horizons, the effect of including the average variable is likely to be greater.

6.5.4 Forecasting scenarios

In real time applications such as short term traffic forecasting, it is desirable to be able to forecast multiple steps into the future in order to provide timely information to travellers. For a road user planning a journey, it is useful to know the likely traffic conditions up to an hour in advance of the intended departure time. Long range forecasts are particularly important in the context of travel time forecasting due to the delay in recording of departure travel times which was discussed in Chapter 4. In this section, the performance of the OLKRR model is assessed in terms of its ability to forecast travel times at four forecast horizons, which are 15, 30, 45 and 60 minutes ahead. The data are not aggregated to achieve these forecasts because this reduces the temporal granularity of the data. Instead, forecasts are made of 5 minute aggregated UTTs, at 3, 6, 9 and 12 steps into the future.

There are two strategies that can be used to carry out this type of forecast, which are iterated and direct. In the first strategy, a one step ahead forecast is made and the forecast point is fed back into the model and used to forecast the subsequent time step. This process is iteratively repeated until the desired number of time steps have been forecast. In the second strategy, forecasts are made directly. This is achieved by sampling the series at the desired temporal interval, and using this series to construct a model for each time interval. For example, given 5 minute averaged data, every second point is used to forecast at the 10

minute interval and every third point to forecast at the 15 minute interval. In this study, the second strategy is used as preliminary testing revealed that direct forecasting was more effective than iterated forecasting in the LOKRR model and the comparison models. As a result, the direct approach is taken here. Given an embedding dimension m and a time delay τ, the embedded, delayed series $\mathbf{x}_t(\tau)$ is defined as:

$$\mathbf{x}_t(\tau) = \left[y_t, y_{t-\tau}, y_{t-2\tau}, \cdots, y_{t-(m-1)\tau}, \hat{\mu}_t\right] \tag{6.32}$$

A separate delayed, embedded series is constructed for each of the forecast intervals.

6.5.5 Performance Measures

Evaluating the performance of a time series model is not straightforward, and there are many indices that may be used. In general, an error index can be either scale dependent or scale free. The RMSE index outlined above is not scale free and cannot be used to compare model performance between series with different scales. In the context of space-time series forecasting, it is important to know how consistent the performance of a model is across the study area. Therefore, the normalised root mean squared error (NRMSE) is used to bring the RMSE values into the same range. The NRMSE is defined as:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{6.33}$$

Where y_{max} and y_{min} are the maximum and minimum values of the series, respectively. RMSE varies with the variability within the distribution of error magnitudes, the square root of the number of errors and the average-error magnitude. It has been argued that an absolute error measure is more appropriate when comparing the performance of models (Willmott and Matsuura, 2005). In this case, the mean absolute percentage error (MAPE) is used as percentages are easy to interpret conceptually. The MAPE is defined as:

$$MAPE = 1/n \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \tag{6.34}$$

The mean absolute scaled error (MASE) is a generally applicable measurement of forecast accuracy (Hyndman and Koehler, 2006). The MASE compares the absolute errors in the forecast values with the absolute errors of a naïve forecast, where the previous value of the series is used to forecast the next value: $\hat{y}_t = y_{t-1}$. As the naïve model is the simplest possible model, any more complicated time series model should outperform it as a minimum requirement. The MASE is defined as:

$$MASE = 1/n \sum_{t=1}^{n} \left(\frac{|e_t|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \right) \tag{6.35}$$

Where e_t is the forecast error for a given period, and the denominator is the average forecast error of the naïve forecast method. A MASE value less than 1 indicates that the forecast model outperforms the naïve model, while a value greater than 1 indicates the opposite. The strength of the MASE is that it brings the concept of forecastability of the data to the performance evaluation. For example, if one achieved a MAPE of 1% on series A with model A and 10% on series B with model B, one may assume model A is a better model than model B. However, if model A achieved a MASE of 1.2 while model B achieved a MASE of 0.8, then model B can be interpreted as the better model. Model A achieves a lower MAPE because series A is more forecastable than series B.

6.5.6 Benchmark Models

In order to place the results in the context of the traffic forecasting literature, the OLKRR model is compared with three benchmark models that have been widely used in the literature, which are ARIMA, ANN and SVR. These models are described in turn below. Only a brief overview of each model is given. Readers are directed to relevant texts in each of the subsections for in depth treatments of the methodologies.

6.5.6.1 ARIMA

The ARIMA model is a linear statistical time series model that has been widely used in many time series forecasting applications, including traffic forecasting. ARIMA is well described in the literature, and readers are directed to the texts of Box et al. (1994) and Hamilton (1994) for detailed treatments of the model. ARIMA is commonly used as a benchmark model against which to judge the performance of times series forecasting models. It is therefore

used here in order to follow convention. An ARIMA(*p,d,q*) model is described by its autoregressive order *p*, its moving average order *q*, and the number of differences *d*. In time series with cyclical trends, a seasonal (S)ARIMA(*p,d,q*)*(*P,D,Q*) model is often used, where *P* is the seasonal autoregressive order, *D* is the order of the seasonal difference, and *Q* is the seasonal moving average order. To maintain consistency with the LOKRR approach, the multi-step forecasts are made in a single step, rather than recursively[23]. As the model building procedure of ARIMA requires a continuous time series, τ embedded, delayed series are constructed and concatenated to form a single series on which the model is trained. For example, consider a time series with its origin at $t = 0$. If $\tau = 3$, three separate series are constructed, one beginning at $t = 0$, one at $t = 1$ and one at $t = 2$, which are merged into a single series.

The function *auto.arima* in the R package *forecast* is used to train the model (Hyndman and Khandakar, 2007). The Akaike Information Criterion (AIC) is used to determine the parameters of the ARIMA models, which is common alternative to the Box-Jenkins procedure (Ozaki, 1977). Nonseasonal and seasonal model specifications are considered in the model building procedure.

6.5.6.2 Support Vector Regression

SVR is a kernel based machine learning method that has demonstrated strong performance in forecasting of traffic variables in recent years (see Chapter 4). An excellent introduction to SVR is given in Smola and Schölkopf (2004). SVR is similar to KRR, but has the advantage of maintaining a sparse representation of the solution. The ε-insensitive variant of SVR (ε- SVR) is used here. There are a minimum of three parameters to train in an ε- SVR model, which are:

1) The kernel parameter(s)
2) The regularisation constant C (similar to λ in kernel ridge regression).
3) The width of the tube ε.

In this case, the same RBF kernel is used as in equation 6.1 and the kernel parameter σ is determined using the same method as the LOKRR model. The median, 0.25 and 0.75 quantiles of the distances between data points are used as the values to maintain consistency with the LOKRR method. The value of C is varied in the range

[23] This approach was found to produce markedly superior results to the iterated approach in preliminary testing

$C = \{0.1,1,10,100\}$. ε is varied in the range $\varepsilon = \{0.0001, 0.001, 0.01, 0.1\}$. After initial feasibility testing, the kernel size is limited to 70 days to ensure training times are acceptable. This means a kernel of size 180*70=12600 is created for each link.

6.5.6.3 Artificial Neural Network

The ANN is the most widely applied traffic forecasting method in the literature. The diverse range of ANNs that have been used for traffic forecasting was revealed in Chapter 3. Unfortunately, the source code for each of these ANN architectures is not readily available, and significant time and effort would be required to implement each of them. Furthermore, as each of the architectures has been developed using a specific dataset, it is difficult to make a reasoned choice about which would make the best comparison. In time series forecasting, ANN models with recurrent architectures have displayed the best results. The Jordan (1986) and Elman (1990) networks are two examples of these. Here, the Elman network is selected as the comparison model as a preliminary analysis revealed superior performance over the Jordan network on this dataset. It must be noted, however, that different ANN structures may yield different results to the Elman network presented here. There is one parameter to be determined in the Elman network, which is the number of input layers. Here, the number is varied between 1 and 10. To train the Elman network, the RSNNS library in R is used, which is based on the Stuttgart Neural Network Simulator (Bergmeir and Benítez, 2012).

6.6 Results

In this section, the forecasting results are presented. First, the training performance and fitted model parameters are examined in section 6.6.1. Following this, in section 6.6.2, the results in the testing phase are presented, including a comparison with the benchmark models.

6.6.1 Training Phase

Table 6.2 shows the training errors and model parameters for each of the links at each of the forecast intervals. As explained in section 6.5.2, RMSE is the error criterion used for model selection. The NRMSE, MAPE and MASE are shown to aid interpretability of the results. The results in terms of NRMSE and MAPE reveal considerable variation in forecast accuracy between links. For instance, at the 15 minute interval, the best MAPE for link 24 is 24.91, whereas the best MAPE for link 1815 is 4.7. This implies that link 1815 is more forecastable

than link 24 and highlights the usefulness of the MASE as an error index[24]. The errors increase as the forecasting interval increases, which is to be expected as there is less local temporal information available at longer forecast horizons. However, even at the 60 minute interval, most of the links exhibit MAPE values of between 10 and 20.

Table 6.2 – Training errors at: a) 15 and 30 minute forecast horizons, and; b) 45 and 60 minute horizons

a)	15 Minute				30 Minute			
Link ID	**RMSE**	**NRMSE**	**MAPE**	**MASE**	**RMSE**	**NRMSE**	**MAPE**	**MASE**
24	0.1081	0.0964	24.91	0.9351	0.1254	0.1119	29.42	0.8869
26	0.0477	0.0690	15.90	0.8302	0.0505	0.0731	16.78	0.7617
442	0.0553	0.0539	13.69	0.9218	0.0665	0.0647	15.93	0.8467
453	0.0385	0.0622	13.41	0.8983	0.0413	0.0668	14.34	0.8304
454	0.1051	0.0366	16.62	0.9311	0.1259	0.0439	19.48	0.8171
881	0.0189	0.0307	11.27	0.8798	0.0222	0.0360	13.16	0.8286
1798	0.0209	0.0350	9.38	0.8412	0.0304	0.0508	13.94	0.7861
1799	0.0257	0.0360	8.58	0.8453	0.0347	0.0487	9.91	0.7602
1815	0.0183	0.0219	4.71	0.9510	0.0334	0.0399	8.35	1.0853
2448	0.0839	0.0607	16.86	0.9334	0.1038	0.0751	20.76	0.8874

b)	45 Minute				60 Minute			
Link ID	**RMSE**	**NRMSE**	**MAPE**	**MASE**	**RMSE**	**NRMSE**	**MAPE**	**MASE**
24	0.1308	0.1167	30.78	0.8343	0.1343	0.1198	32.14	0.8007
26	0.0516	0.0747	17.16	0.7226	0.0526	0.0761	17.62	0.6937
442	0.0745	0.0725	17.82	0.8315	0.0811	0.0790	18.94	0.7990
453	0.0420	0.0679	15.13	0.7883	0.0484	0.0524	16.01	0.7721
454	0.1379	0.0481	20.75	0.7417	0.1461	0.0509	21.17	0.7023
881	0.0260	0.0422	14.38	0.7743	0.0276	0.0448	15.03	0.7479
1798	0.0349	0.0584	17.16	0.7451	0.0388	0.0650	19.56	0.6938
1799	0.0401	0.0563	10.60	0.6950	0.0421	0.0592	11.04	0.6528
1815	0.0389	0.0464	10.62	1.0491	0.0437	0.0522	12.38	1.0305
2448	0.1162	0.0841	23.66	0.8637	0.1241	0.0897	25.28	0.8319

[24] In fact, link 1815 is not more forecastable, as is demonstrated in section 6.6.1.1

Examining the performance in terms of MASE, it can be seen that all of the models meet the minimum requirement of outperforming the naïve model at each of the forecast horizons, with the exception of link 1815. In general, the performance of LOKRR in terms of MASE improves as the forecasting horizon increases. This implies that the model is successful in capturing the cyclic pattern in the data, which is not captured by the naïve model at increasing forecast horizons.

6.6.1.1 Investigation of model parameters

Table 6.3 summarizes the best parameters for each link and forecast horizon. All of the models exhibit best performance with $\lambda = 1/8\lambda_0$. This suggests that the method outlined in section 6.4.5.1 for choosing λ may overestimate the noise in the data, and that smaller values of λ could be used (for example: $\lambda = \frac{1}{16\lambda_0}, \frac{1}{32\lambda_0}$). The best values of σ vary between the 0.25, 0.5 and 0.75 quantiles of $\|\mathrm{x} - \mathrm{x}'\|^2$. As the forecasting horizon increases, there is a tendency towards larger kernel bandwidths. This reflects the fact that the uncertainty in the forecasts increases with forecasting horizon, so the models tend towards the historical average. With sufficiently large σ, each forecast would reduce to the average of the observations contained within *w*.

Table 6.3 – Fitted model parameters at each of the forecast horizons

	15 minute			30 minute			45 minute			60 minute		
Link	**σ**	**λ**	*w*	**σ**	**λ**	*w*	**σ**	**λ**	*w*	**σ**	**λ**	*w*
24	0.75	1/8	3	0.75	1/8	3	0.5	1/8	1	0.75	1/8	2
26	0.5	1/8	2	0.5	1/8	2	0.5	1/8	2	0.5	1/8	3
442	0.5	1/8	3	0.5	1/8	3	0.75	1/8	3	0.75	1/8	3
453	0.75	1/8	3	0.75	1/8	3	0.75	1/8	3	0.75	1/8	3
454	0.5	1/8	3	0.75	1/8	2	0.75	1/8	3	0.75	1/8	3
881	0.5	1/8	3	0.5	1/8	3	0.75	1/8	3	0.75	1/8	3
1798	0.5	1/8	1	0.5	1/8	2	0.75	1/8	3	0.75	1/8	3
1799	0.5	1/8	3	0.75	1/8	3	0.75	1/8	3	0.75	1/8	3
1815	0.5	1/8	3	0.5	1/8	2	0.5	1/8	2	0.5	1/8	2
2448	0.25	1/8	2	0.5	1/8	2	0.75	1/8	3	0.75	1/8	3

Note: σ values are quantiles of $\|\mathrm{x} - \mathrm{x}'\|^2$; λ values are multiples of λ_0

The model parameters σ and λ of the LOKRR are local and vary throughout the day. Each time point has a different value of σ and λ which is determined from the data. Figure 6.5 shows the fitted values of the 0.25, 0.5 and 0.75 quantiles of σ at each time point and each link.

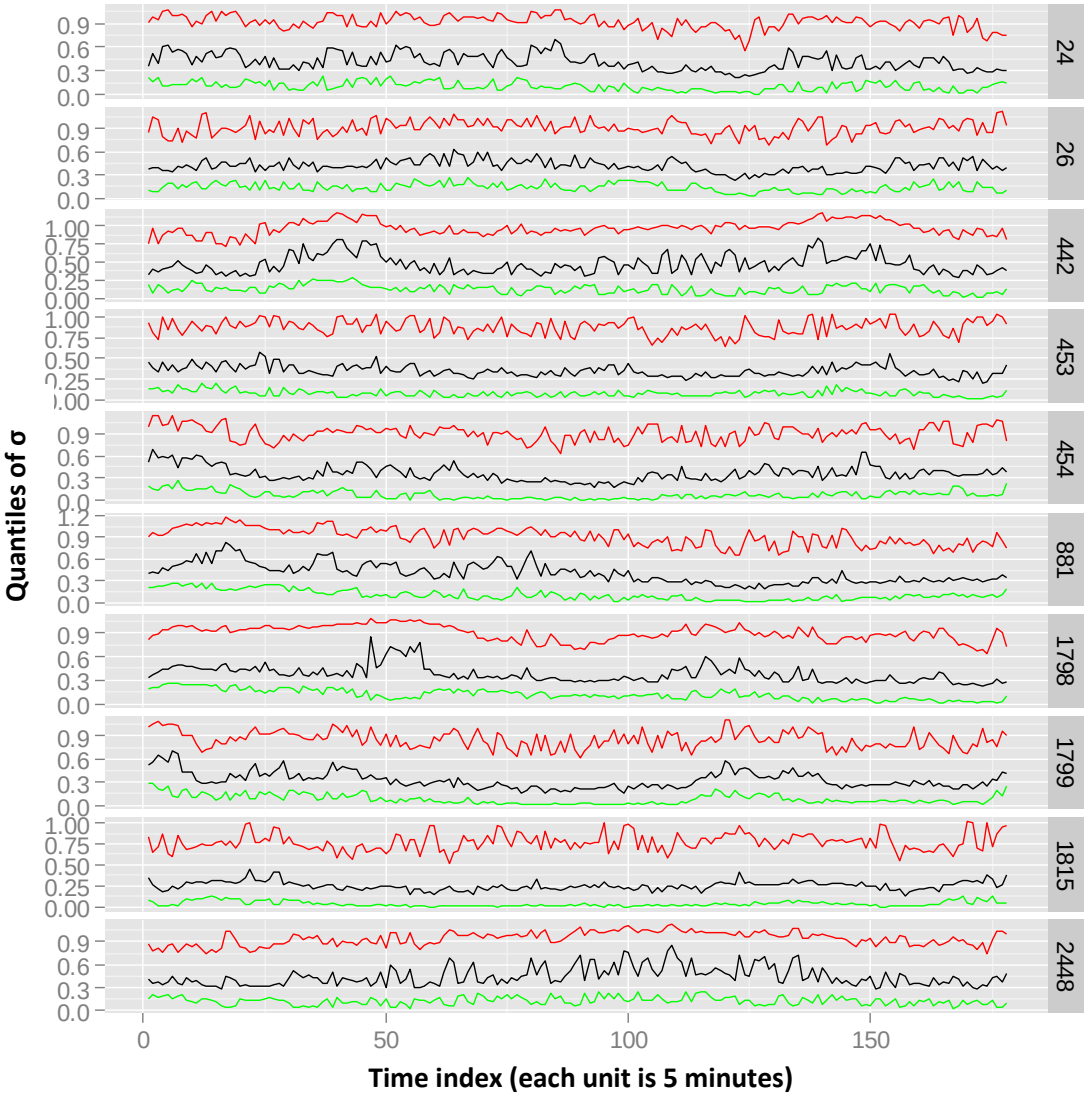


Figure 6.5 – Fitted values of sigma with a window size of 1. Red, black and green lines are the 0.75 quantile, median, and 0.25 quantiles respectively.

The median value of σ tends to be higher in the morning and evening peak periods than other times of day. This is particularly evident on links 442, 1798 and 1799. In general, σ is not static in time, which reflects the fact that the distribution of the data is conditional on the time of day. The finding of larger values of σ in the peak periods is consistent with empirical studies of traffic data, which have demonstrated that the mean/median and variance/interquartile range of travel times is higher in peak periods (e.g. Fosgerau and

Fukuda, 2012). However, not all of the links exhibit the same two peak profile. This is because the traffic patterns of urban traffic links depend on many factors, including whether they are arterial (inbound or outbound) or radial routes. It is important that a model is able to capture these differences.

Figure 6.6 shows the average value of λ_0 by time of day for each of the test links. The estimated values of λ_0 are higher around the peak periods. Recalling that the values of λ_0 are obtained from the signal to noise ratio φ derived from the OLS fit of the data (see section 6.4.5.1), this result indicates that there is a higher level of noise in the data around the peak periods. This reflects the basic intuition that traffic conditions are more difficult to predict in peak times. This is particularly marked on link 1815, which is characterised by rare, but very extreme peaks in the morning rush hour period. In this case, the value of λ_0 rises to around 80 because OLS cannot model the highly nonlinear input output relationship in this period. The variation in travel times is interpreted as noise. The implication of the higher values of λ_0 in the peak periods is that forecasts will tend to be closer to the average because high values of λ penalise large weights. This is undesirable when forecasting non-recurrent congestion events. However, searching the grid of multiples of λ_0 in a cross-validation approach circumvents this problem. On other links, the values of λ_0 are higher during the early morning and late evening periods. In these cases, the parameter values correctly identify the increased noise in the data at times of lower flow.

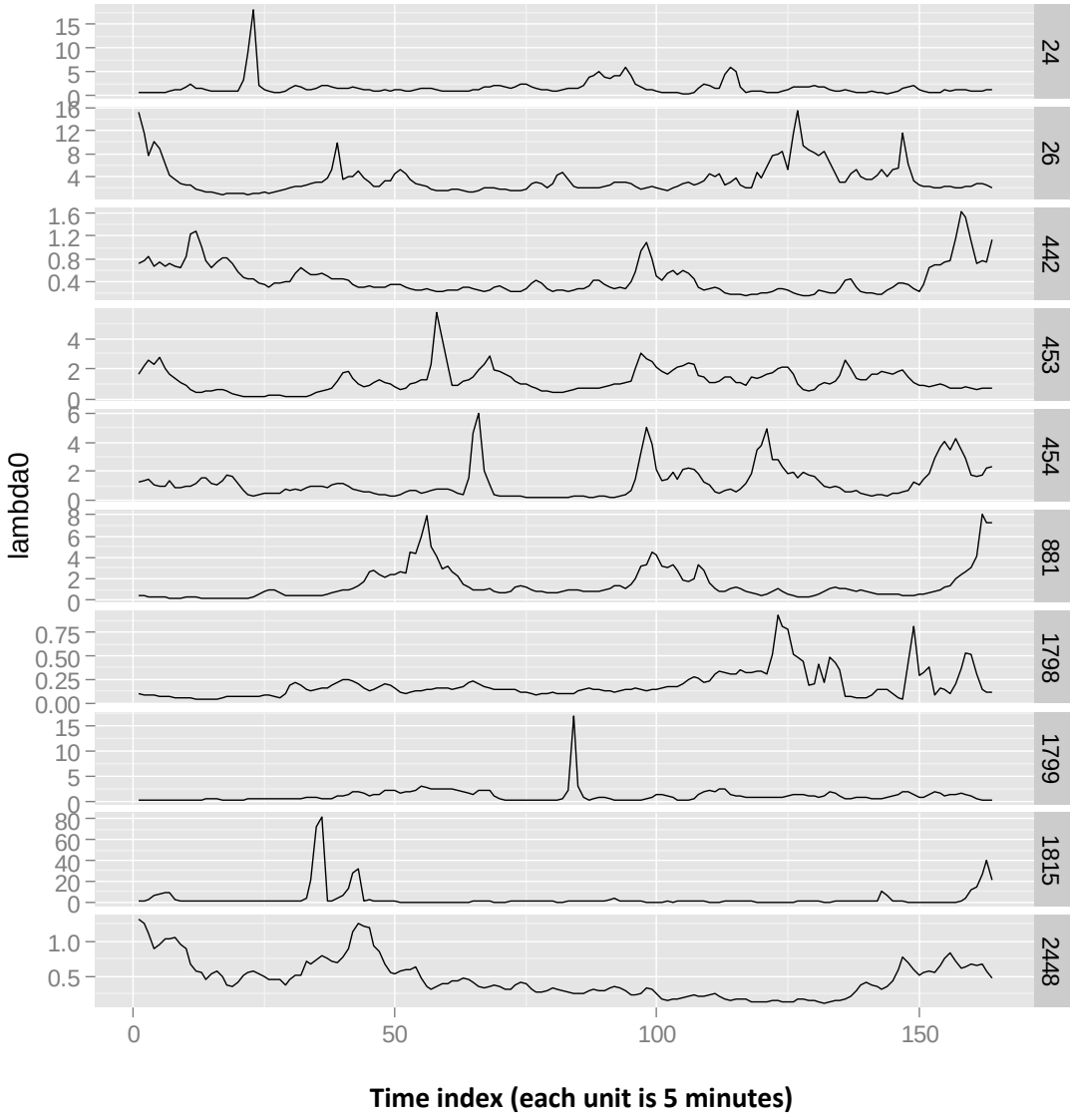


Figure 6.6 – Fitted values of Lambda 0 with w=1

6.6.1.2 Sensitivity of model parameters

In this section, the sensitivity of the model performance with respect to the parameters is investigated. In this context, the meaning of parameter sensitivity is the effect that a change in the value of a parameter has on the performance of the model. As mentioned in section 6.4.5, there is an extensive literature on parameter selection in kernel methods such as SVR and KRR. However, in practice one may be willing to accept a suboptimal model if the advantage gained in training time reduction or ease of implementation outweighs the potential advantage of increased accuracy.

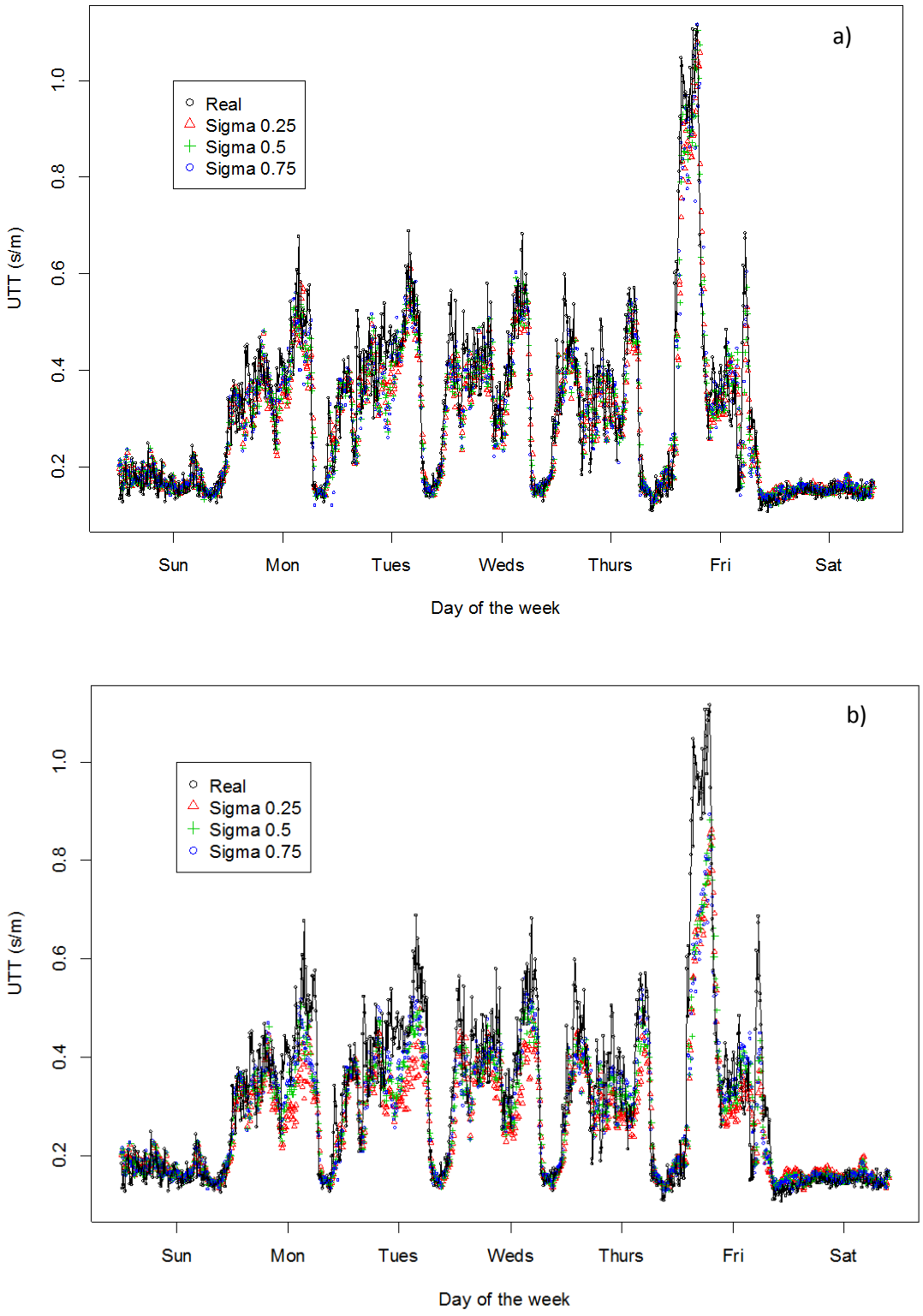


Figure 6.7 – Effect of varying σ through its range with Lambda fixed at: a) 1/8λ_0, and; b) 2λ_0

Figure 6.7 shows the effect of varying σ through its range, while keeping λ fixed in forecasting situation. In the figures, the large peak in UTT on the Friday represents a non-

recurrent congestion event. The effect of varying λ on model performance is significant. Choosing a large value of λ results in an overestimation of the noise in the data and an overly smooth solution. Consequently, models using large values of λ do not perform strongly in forecasting abnormally high peaks in the data. This result, combined with the model parameters of the fitted models shown in table 6.3, indicates that it is not necessary to test larger values of λ, supporting the intuiution of Exterkate (2013, p.6) that λ should be less than $\lambda_0$because one expects to obtain a better fit using nonlinear models.

Figure 6.8 shows the effect of holding σ fixed and varying λ. Figures 6.8 a)-c) all appear broadly similar, indicating that the effect of varying σ is not as great as the effect of varying λ. This supports the statement of Caputo et al. (2002) that any kernel bandwidth between the 0.1 and 0.9 quantiles of the kernel. The implication of this is that the performance of LOKRR is relatively insensitive to Gaussian kernel bandwidth choice when the method described in section 6.5.4.2 is used. Similar results are found in the other links in the network.

Some conclusions can be drawn from the analysis in this section that make model selection more straightforward. Firstly, small values of λ always yield greater accuracy than large values, meaning it is unnecessary to test large multiples of λ_0. Secondly, although performance varies with choice of σ, a good fit to the data can be obtained with the limited parameter range suggested. The latter statement is verified in the comparison with the benchmark models in section 6.6.2.2.

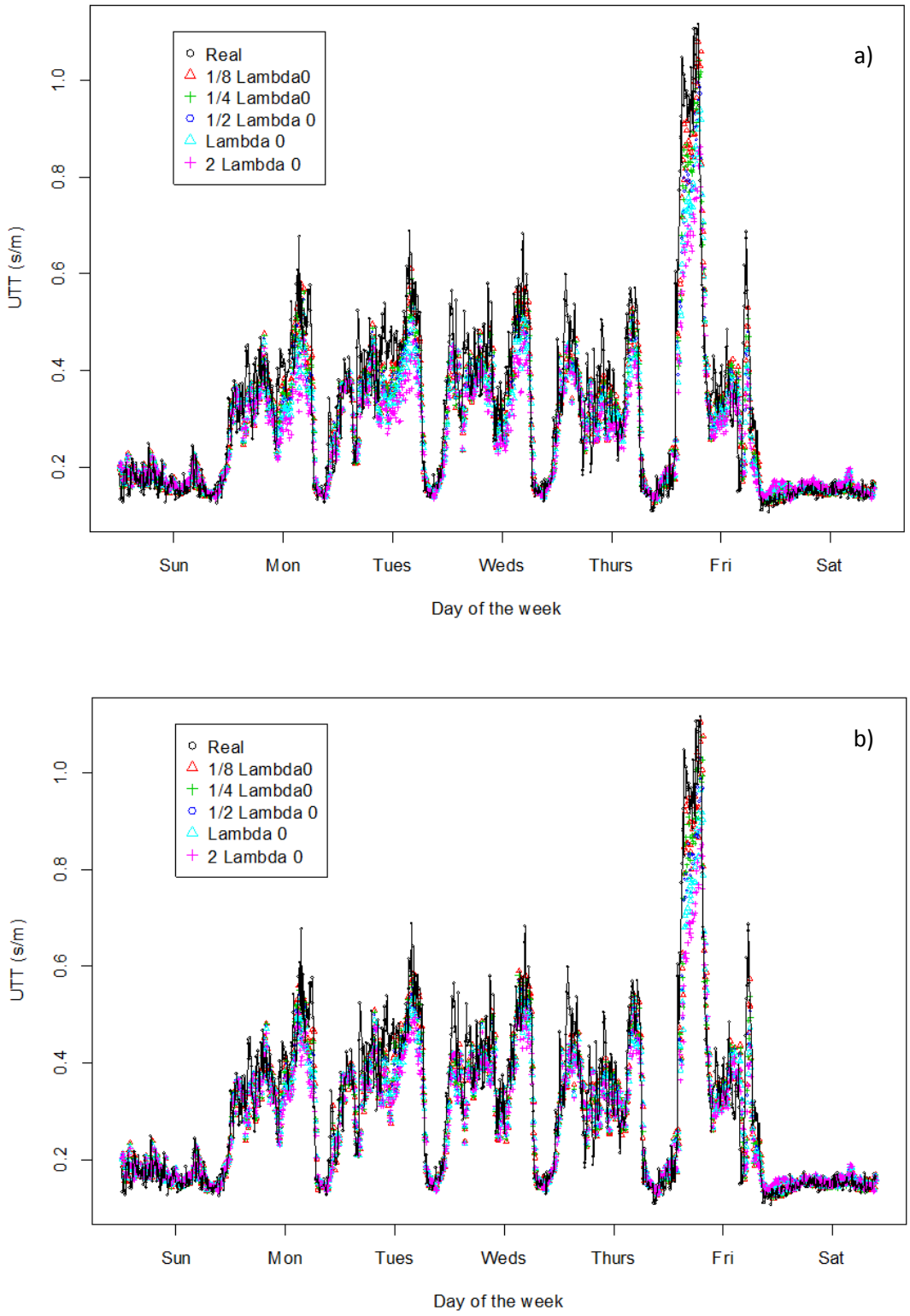


Figure 6.8 – Effect of varying λ through its range with σ fixed at: a) 0.25 quantile, b) median

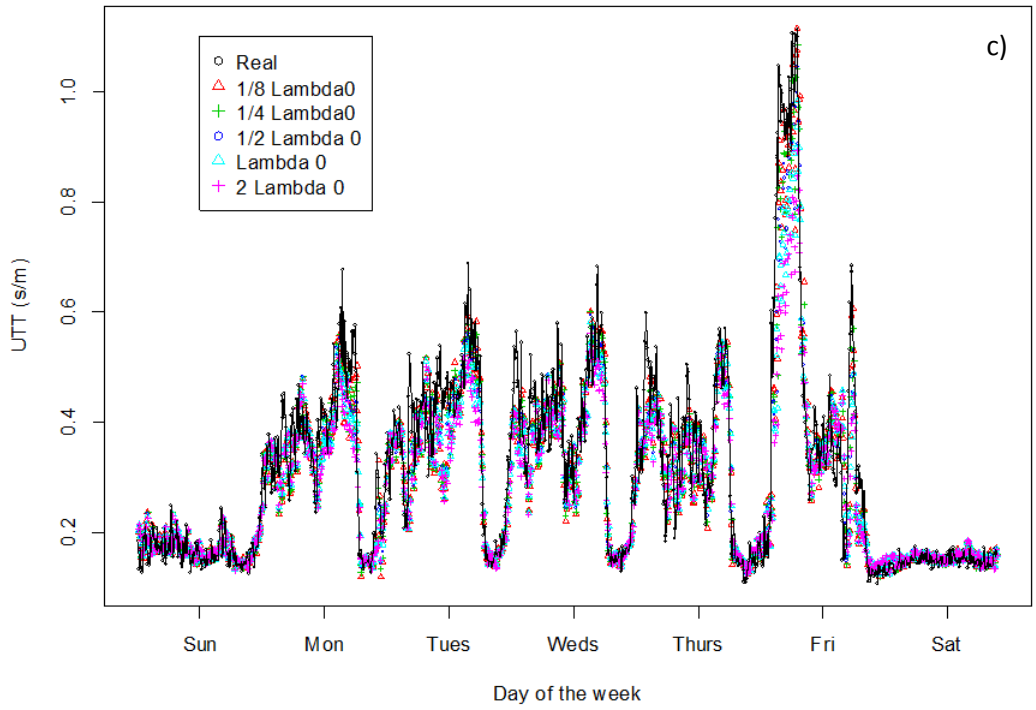


Figure 6.8 (continued) – Effect of varying λ through its range with σ fixed at: c)0.75 quantile

6.6.2 Testing Phase

The parameters obtained in the training phase are used to forecast the remaining 37 days in the validation set. Table 6.4 shows the testing results. In terms of RMSE, the training and testing errors are broadly similar. At the 15, 30 and 60 minute intervals, testing performance is better than training performance on 4 of the 10 links. At the 45 minute interval, testing performance is better than training performance on 5 of the links. The reason for the differences in the training and testing errors may reflect the relative forecastability of the data in each period. The MASE gives a good indication of performance in this regard. For each of the links, the MASE is <1, with the exception of link 1815, which is discussed in the following section. The pattern of decreasing MAPE with increasing forecasting interval observed in the training phase is repeated in the testing phase. These results indicate that the model generalises well to the unseen validation data.

Table 6.4 – Testing Errors of the LOKRR model

15 minute				
	RMSE	**NRMSE**	**MAPE**	**MASE**
1798	0.015056	0.061842	9.896842	0.86972
1799	0.020865	0.041542	9.04533	0.86287
1815	0.021719	0.028766	5.557989	1.066146
2448	0.078816	0.063965	17.51586	0.951652
24	0.135792	0.050655	23.8859	0.887596
26	0.056001	0.050071	16.67646	0.838566
442	0.077271	0.026585	16.10907	0.900641
453	0.05	0.024215	12.96563	0.842421
454	0.083559	0.052184	16.19808	0.906937
881	0.027327	0.030507	12.72943	0.872443
30 minute				
1798	0.021784	0.089477	14.50633	0.778719
1799	0.025402	0.050759	10.22777	0.789565
1815	0.0303	0.040132	8.325483	1.079486
2448	0.104256	0.084611	21.61859	0.897533
24	0.155577	0.058035	28.45854	0.843025
26	0.062764	0.056117	17.64926	0.756886
442	0.095284	0.032783	19.62794	0.832451
453	0.053543	0.025931	14.08368	0.77528
454	0.112563	0.070298	20.42523	0.841052
881	0.036773	0.041052	14.75696	0.817719
45 minute				
1798	0.02598	0.111417	18.02658	0.72652
1799	0.026217	0.052198	11.11021	0.742244
1815	0.035621	0.04718	11.29372	1.116573
2448	0.117528	0.095382	24.21025	0.869196
24	0.168476	0.062847	31.1086	0.825435
26	0.066129	0.059126	18.19328	0.70734
442	0.09972	0.034309	21.12433	0.751533
453	0.054154	0.026227	14.58123	0.742946
454	0.124311	0.077635	23.43005	0.795166
881	0.042819	0.047802	15.85057	0.756781

60 minute				
1798	0.028534	0.117203	20.60327	0.671616
1799	0.025176	0.062291	11.66314	0.735095
1815	0.037106	0.049147	12.53772	1.061102
2448	0.124704	0.101206	25.82053	0.833048
24	0.17265	0.064404	32.58387	0.781458
26	0.066611	0.059556	18.4175	0.6703
442	0.103471	0.0356	22.76551	0.720822
453	0.055945	0.027094	15.38157	0.738591
454	0.130278	0.081362	25.11916	0.768823
881	0.046543	0.051959	17.09304	0.721626

6.6.2.1 Visual analysis of forecasts

In this section, three links are selected for visual analysis of the forecasts, which are links 442, 1798 and 1815. These links are selected to demonstrate the strengths and weaknesses of the LOKRR model. In the former two cases, the model performs strongly in forecasting up to 60 minutes ahead. In the latter case, the model is unable to produce accurate forecasts as the traffic series of the link in question displays nonseasonal characteristics.

Figures 6.9 a) to d) show the observed (blue lines) versus forecast (black dots) values for link 442 at the 15, 30, 45 and 60 minute horizons respectively, for a single week (between days 10 and 17) of the testing period. The historical average UTT over the training period is also shown (grey line) to highlight the deviation from typical conditions. At the 15 minute horizon, the LOKRR model is able to forecast the variation in UTTs quite accurately. On those links that display more pronounced seasonal patterns, the model is able to forecast the onset of the peak periods without a time delay. This is because the model structure captures the cyclic nature of the data. However, in some cases where there are sudden increases in travel times, the forecast values lag behind the real values. This is to be expected as it is not possible to forecast changes in traffic condition that occur after the most recently observed traffic pattern.

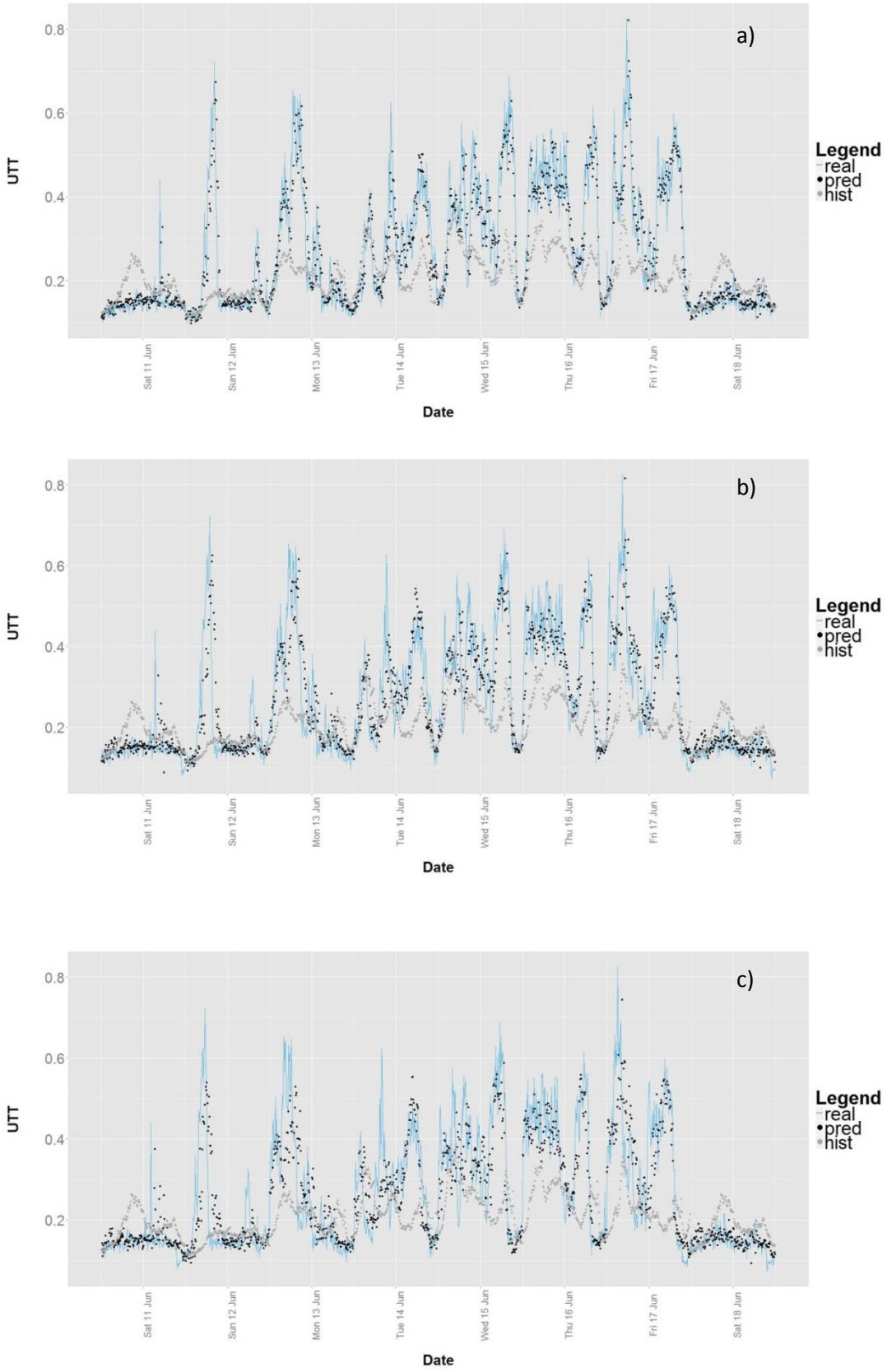


Figure 6.9 – Observed versus forecast values for link 442 at a) 15 minute; b) 30 minute; c) 45 minute intervals.

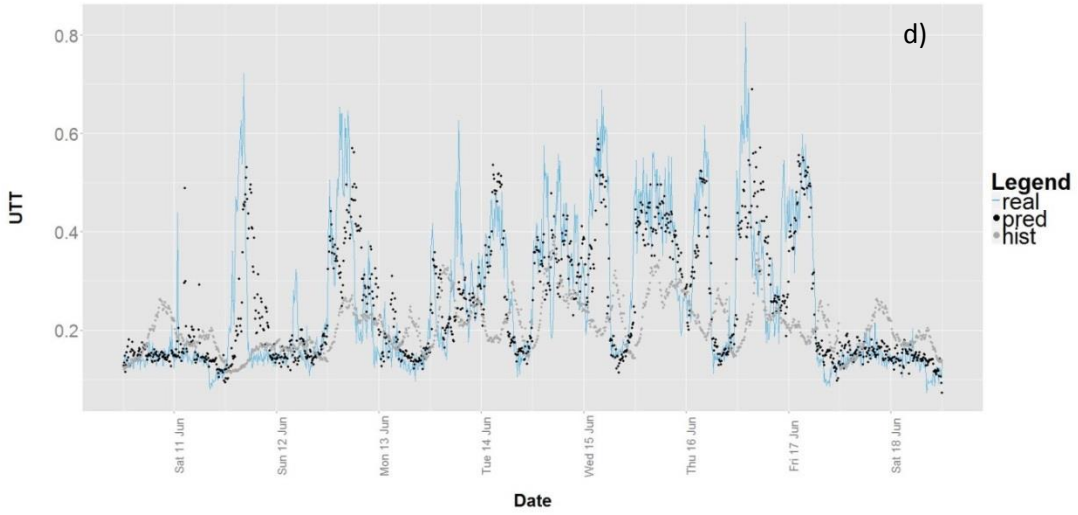


Figure 6.9 (continued) – Observed versus forecast values for link 442 at d) the 60 minute interval.

As the forecasting horizon increases, the ability of the model to capture the atypical traffic conditions diminishes. Again, this is to be expected as the traffic situation can change substantially over the course of 30 minutes to one hour. However, the model still performs reasonably well in capturing the variation around the recurrent traffic patterns.

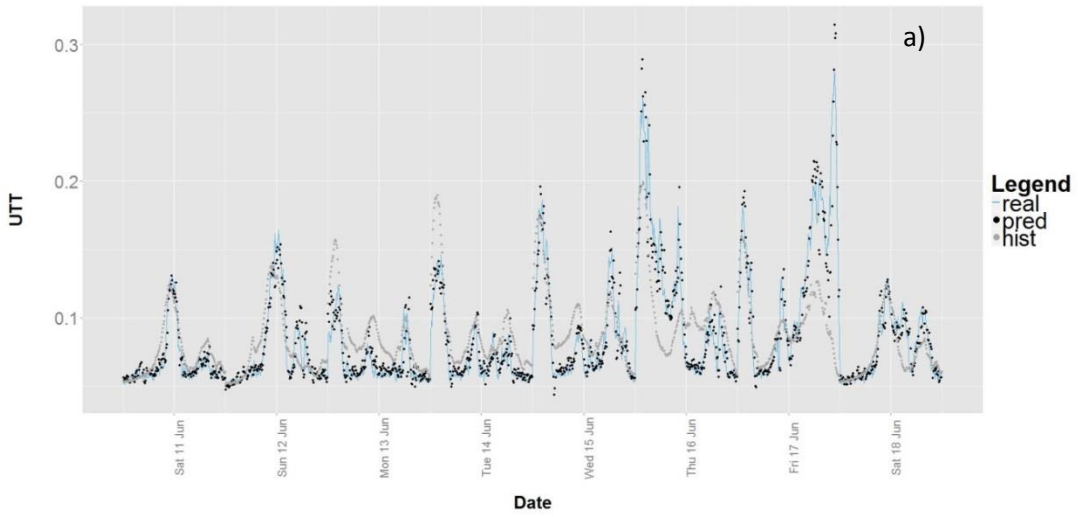


Figure 6.10 – Observed versus forecast values for link 1798 at a) 15 minute interval

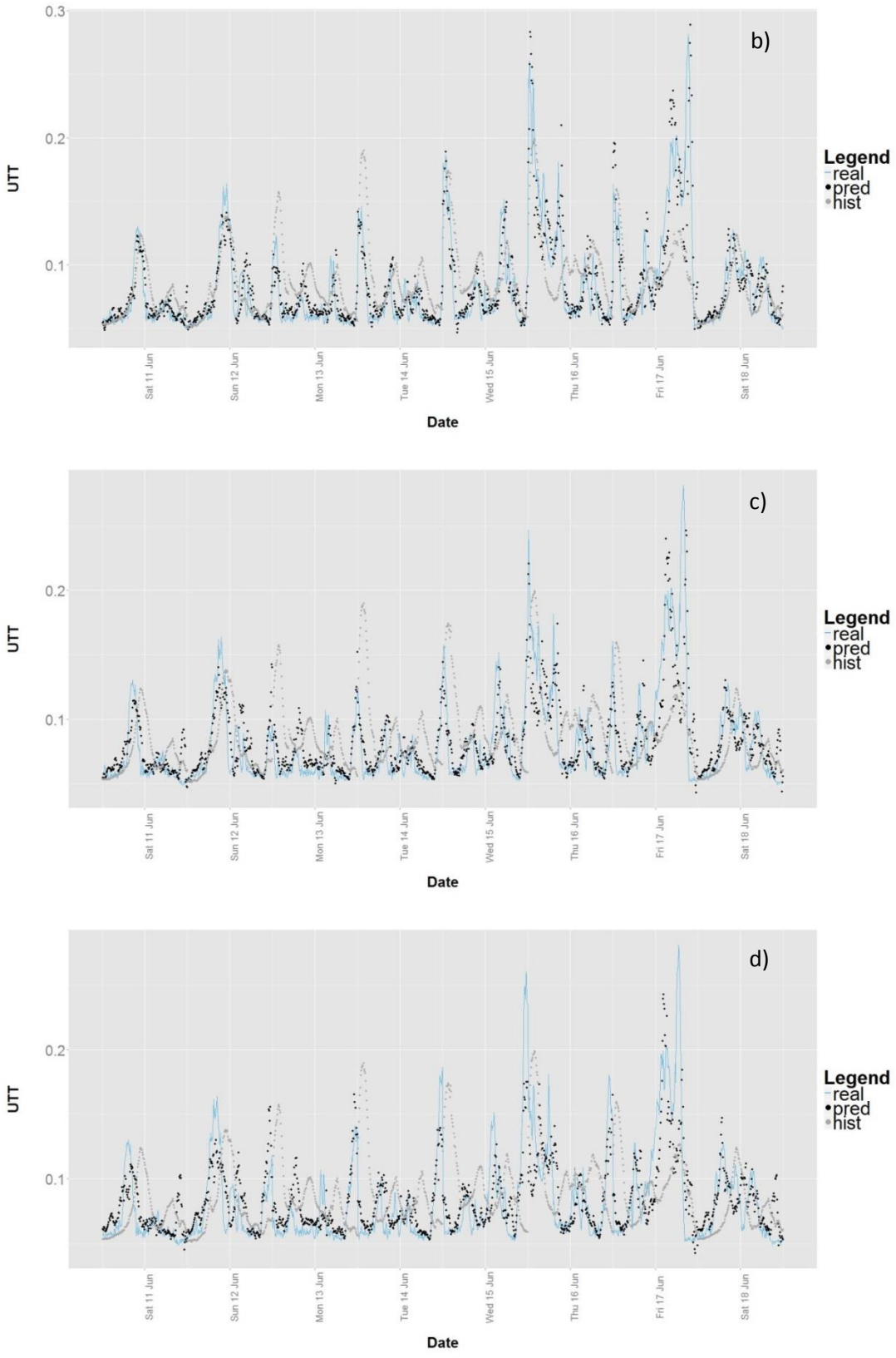


Figure 6.10 (continued) – Observed versus forecast values for link 1798 ; b) 30 minute; c) 45 minute intervals; d) 60 minute intervals

Figure 6.10 shows the observed versus forecast values at each of the forecast intervals for link 1798. The model displays similar performance on this link to link 442. At the 15 minute horizon, the model is able to forecast the onset, dissipation and severity of congestion with good accuracy. The effectiveness of the model is confirmed by comparison with the historical average over the training data. As the forecast horizon increases, the lag between the observed data and the forecasts is again apparent, but the recurrent onset of congestion is still accurately forecast without a time delay. These results reflects the typical model performance.

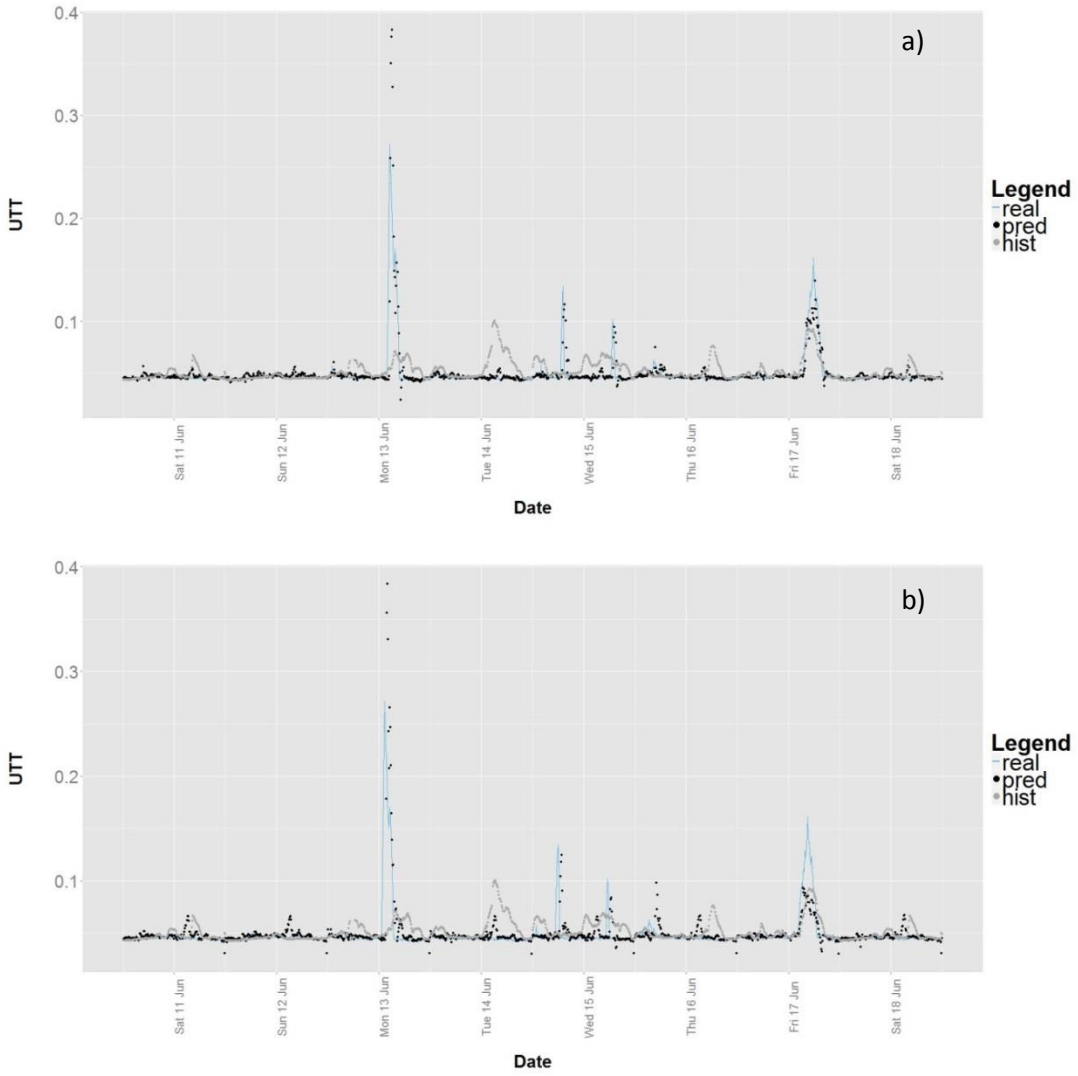


Figure 6.11 – Observed versus forecast values for link 1815 at a) 15 minute; b) 30 minute intervals

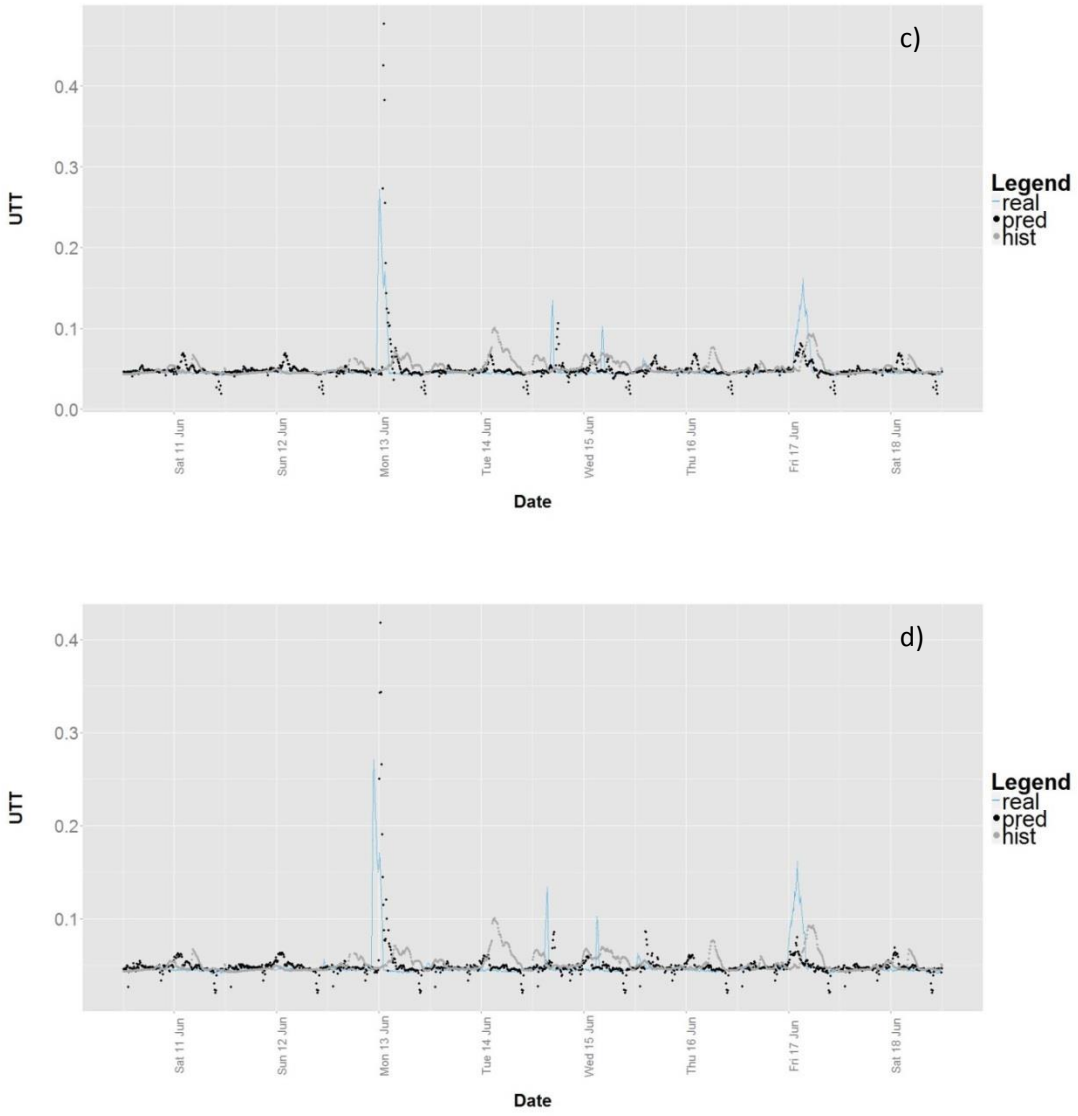


Figure 6.11 (continued) – Observed versus forecast values for link 1815 at c) 45 minute and d) 60 minute interval

Figure 6.11 shows the observed versus forecast values for each of the forecast intervals on link 1815. Link 1815 is characterised by low UTTs interrupted by occasional severe congestion events. Although the model is able to forecast the large peak on Monday 13th June at the 15 minute interval to an extent, its magnitude is exaggerated. The performance on the peaks also comes at the expense of performance under recurrent conditions. Furthermore, as the forecast interval increases, there is a lag in the forecast equal to the forecast horizon. Because the duration of the congestion event is approximately one hour, the model forecasts its onset after traffic conditions have returned to normal levels. Link 1815 highlights a limitation of the methodology. The LOKRR model in its present form is

designed to model statistical properties of the data that vary smoothly with time. In the case of link 1815, the traffic can be seen as operating at multiple scales at the same times of day. This type of behaviour cannot be modelled effectively using a single kernel bandwidth for each time of day. Some suggestions for dealing with this type of traffic behaviour are given in Chapter 8.

6.6.2.2 Comparison with benchmark models

In this section, the empirical performance of the LOKRR model is compared with the performance of the benchmark models that were introduced in section 6.5.6. Table 6.5 shows the average errors of the LOKRR model and the comparison models in the training and testing periods at each of the forecast intervals. In terms of RMSE, the LOKRR model is the best performing model in both the training and testing phases at each forecast horizon.

Table 6.5 – Comparison with benchmark models – average errors at: a) 15 minute; b) 30 minute; c) 45 minute, and; d) 60 minute forecast horizons

a)	Training			Testing		
Model	RMSE	NRMSE	MAPE	RMSE	NRMSE	MAPE
LOKRR	0.051578	0.049348	13.24606	0.056641	0.043033	14.05806
SVR	0.063477	0.060735	13.35697	0.067139	0.050494	14.16037
ANN	0.062049	0.059943	14.58549	0.062892	0.056895	14.91349
ARIMA	-	-	-	0.057455	0.048411	14.89455

b)	Training			Testing		
Model	RMSE	NRMSE	MAPE	RMSE	NRMSE	MAPE
LOKRR	0.062015	0.059734	15.67754	0.069825	0.05492	16.96798
SVR	0.076575	0.073219	16.06394	0.081147	0.062619	17.26276
ANN	0.077809	0.075266	17.25284	0.081186	0.071484	18.05018
ARIMA	-	-	-	0.074985	0.065109	18.90466

c)	Training			Testing		
Model	RMSE	NRMSE	MAPE	RMSE	NRMSE	MAPE
Online KRR	0.067464	0.064846	17.27787	0.076096	0.061412	18.89288
SVR	0.083118	0.079573	17.7488	0.088623	0.069456	19.18715
ANN	0.085629	0.083494	18.49986	0.089811	0.070026	19.29111
ARIMA	-	-	-	0.090186	0.074243	21.68558

d)	Training			Testing		
Model	RMSE	NRMSE	MAPE	RMSE	NRMSE	MAPE
LOKRR	0.071581	0.066783	18.40614	0.079102	0.064982	20.19853
SVR	0.088056	0.081424	18.83573	0.093475	0.074376	20.17787
ANN	0.092487	0.090328	19.94198	0.097288	0.074856	21.07094
ARIMA	-	-	-	0.089645	0.07822	23.0101

Note: Training errors are not shown for the ARIMA model due to the difference in the model fitting procedure.

In terms of MAPE, LOKRR outperforms each of the comparison models at the 15 minute, 30 minute and 45 minute horizons. The SVR model performs marginally better at the 60 minute horizon in terms of average MAPE. Of the comparison models, ARIMA performs best at shorter forecast horizons, but its performance decreases rapidly as the forecast horizon increases. The ANN and SVR models outperform ARIMA at the 45 minute and 60 minute horizons.

Figure 6.12 shows the RMSE of each of the models at the four forecasting horizons. LOKRR is the best performing model in terms of RMSE on 5, 9, 9 and 10 of the links at the 15, 30, 45 and 60 minute forecasting horizons respectively, highlighting the consistency in the performance of the model. The ARIMA model exhibits the best performance on 4, 1 and 1 of the 10 links at the 15, 30 and 45 minute forecast horizons respectively. The SVR and ANN models exhibit best performance on a single model each.

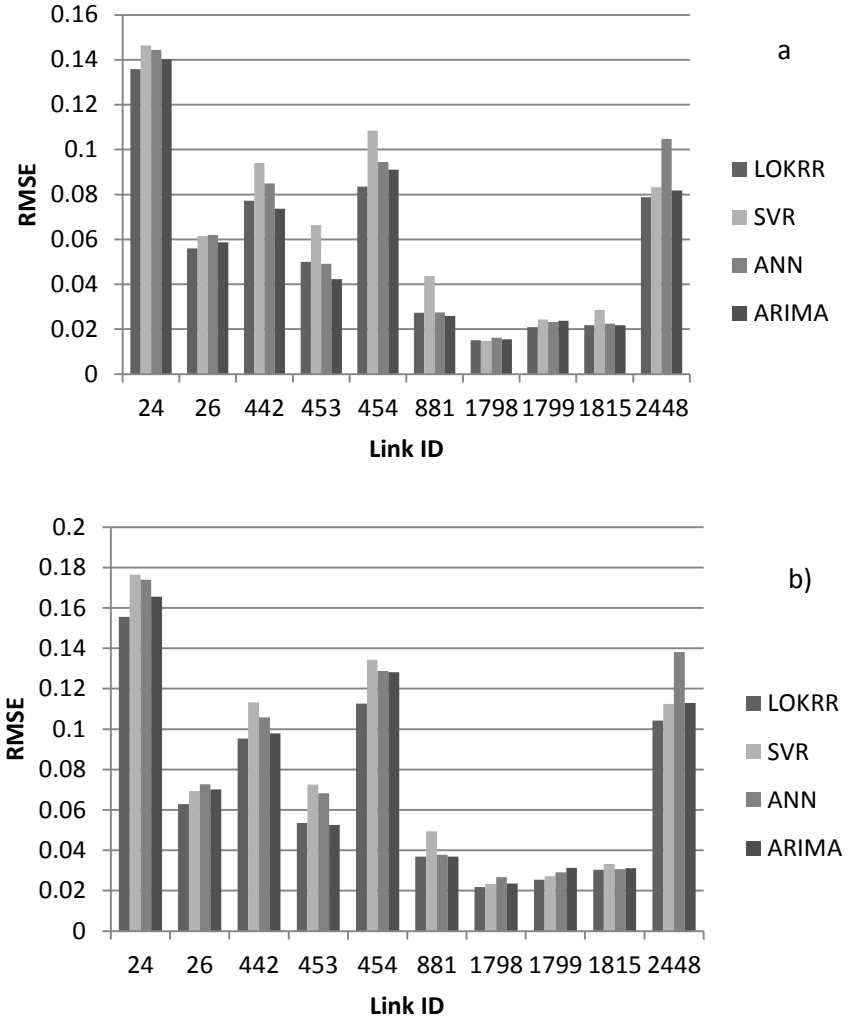


Figure 6.12 – RMSE of each of the models at: a) 15 minute; b) 30 minute forecast horizons

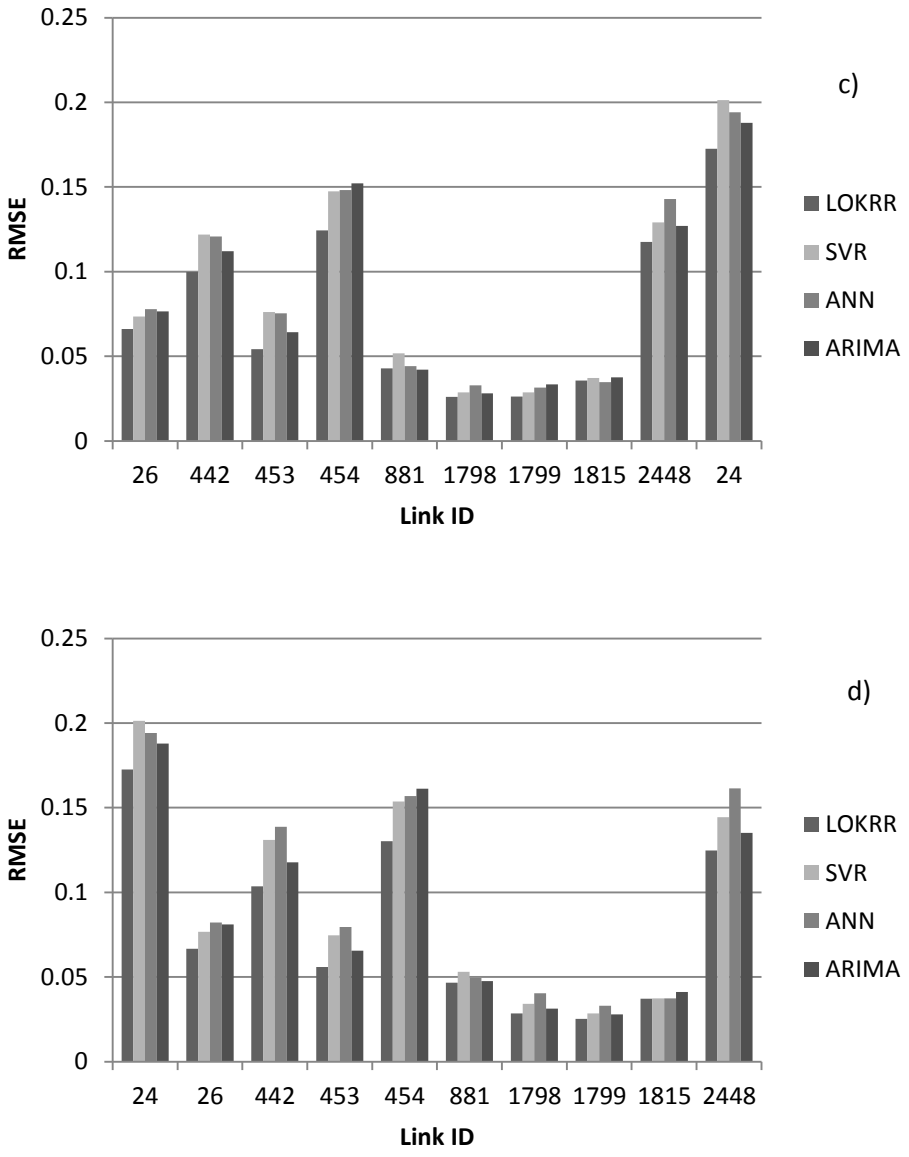


Figure 6.12 (continued) – RMSE of each of the models at: c) 45 minute, and; d) 60 minute forecast horizons

In terms of MAPE, LOKRR is the best performing model on 7, 6, 6 and 5 of the links at the 15, 30, 45 and 60 minute horizons respectively. SVR also performs strongly, achieving the lowest MAPE on 3, 3, 1 and 3 of the links at the 15, 30, 45 and 60 minute horizons respectively. ANN is the best performing model on 1, 3 and 2 of the links at the 30, 45 and 60 minute intervals respectively. The ARIMA model does not perform well in terms of MAPE, and is not the best performing model in any of the cases.

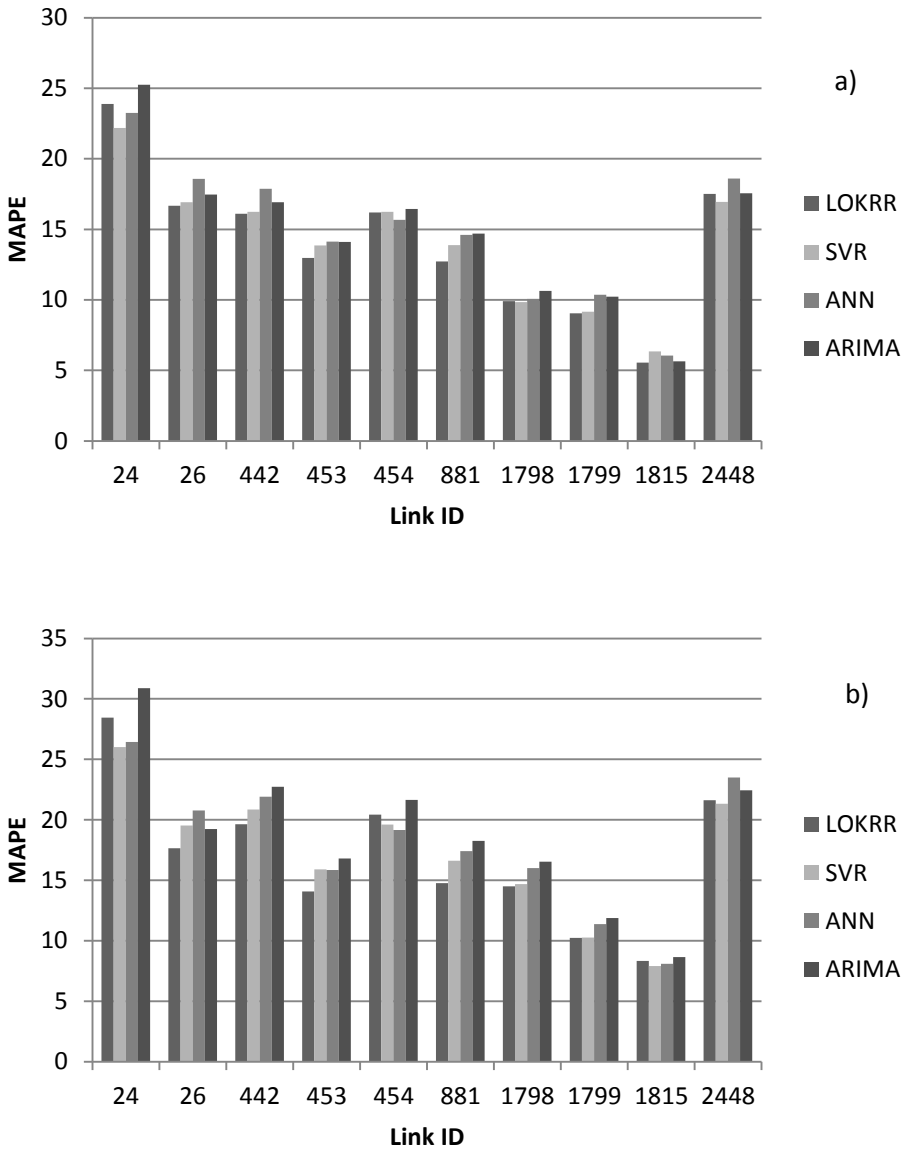


Figure 6.13 – MAPE of each of the models at: a) 15 minute; b) 30 minute forecast horizons

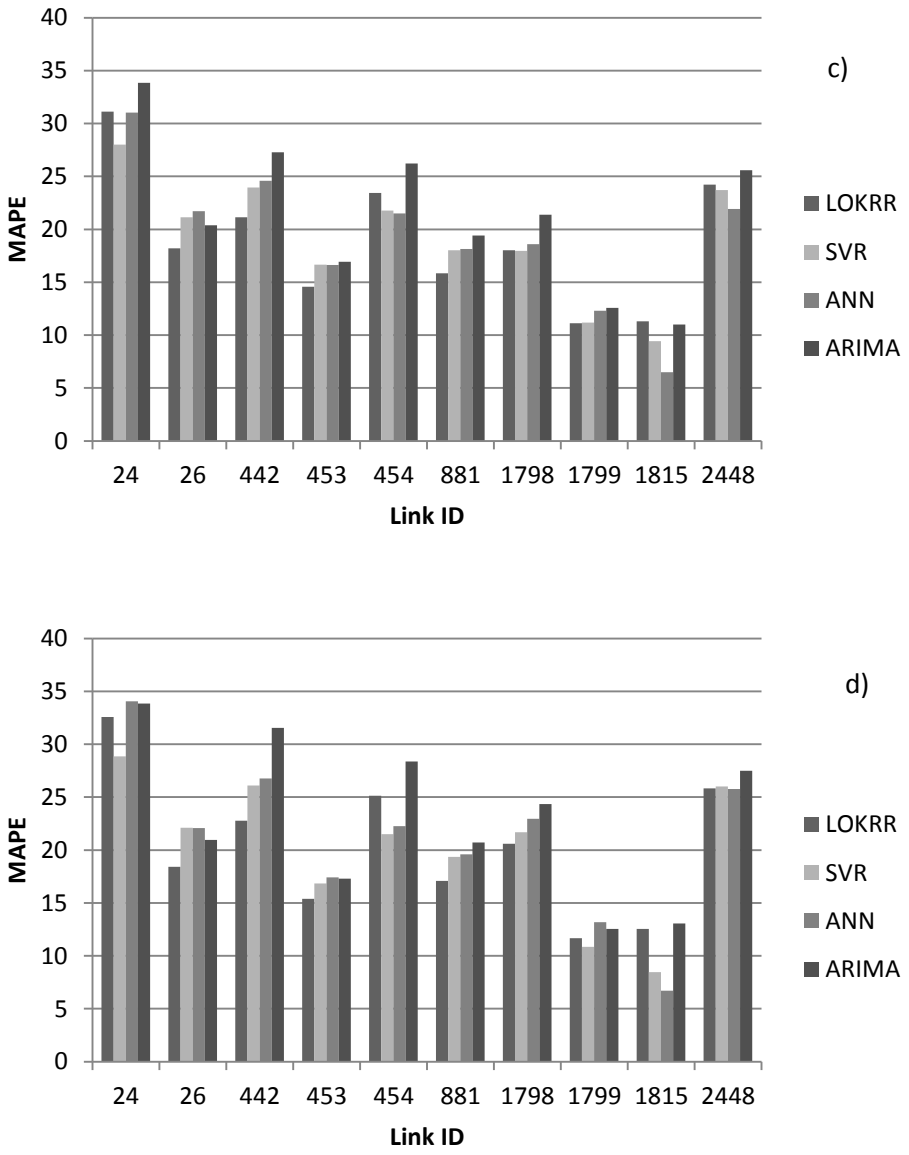


Figure 6.13 (continued) – MAPE of each of the models at: c) 45 minute, and; d) 60 minute forecast horizons

6.7 Chapter Summary

In this chapter, a temporally local forecasting model, namely LOKRR, has been described for forecasting seasonal time series. To incorporate seasonality, a separate kernel is defined for each time point with its own parameters. This structure has two main benefits. Firstly, it enables smaller kernels to be accessed each time a forecast is made, making the algorithm efficient. Secondly, it allows the model to capture the seasonality and heteroskedasticity in the series. The empirical case study demonstrates the strong performance of the model in forecasting up to one hour ahead. This ability makes the model suitable for real time

application, where accurate forecasts of future traffic conditions are important. However, situations where the model is not effective are revealed, and some suggestions are made for addressing this in future research.

The model is compared with three commonly used benchmark algorithms; ARIMA, SVR and Elman ANN. Performance is at least comparable with each of these models, and superior in the majority of cases, at each of the forecast horizons.

The strength of the model described here lies in its ability to model the cyclic variation in travel times throughout the day. Each forecast is a weighted nonlinear combination of previously observed travel times from within the temporal window *w*. This constrains the model to only produce forecasts based on what has happened during this window in the past. Therefore, it naturally captures the cyclic pattern in the data. At shorter forecasting intervals, the model is able to use recent traffic patterns to accurately forecast the variation above or below the average conditions. As the forecast horizon increases and more uncertainty is introduced to the forecasts, the model tends towards the historical average. However, it is still able to produce more accurate forecasts than the historical average, indicating that it captures the local conditions even at longer forecast horizons.

Referring back to the model requirements stated in Chapter 5.5. The model described in this Chapter has attempted to address the generic requirements of: 1) accuracy and reliability; 2) forecasting multiple time steps into the future so as to be fit for purpose; 3) adapting to changes in the distribution of the data. It has also attempted to address the application specific requirements of forecasting up to 1 hour ahead for ATIS. However, as LOKRR is a purely temporal model, it cannot deal with the model requirements in terms of dealing with missing and corrupt data in real time and being applied to large networks. In the following Chapter, the LOKRR model is extended to incorporate the spatial structure of networks in order to address the remaining model requirements.

Chapter 7 - Spatio-temporal LOKRR for forecasting under missing data

7.1 Chapter Overview

In this Chapter, the LOKRR model described in Chapter 6 is extended to accommodate spatio-temporal structure. The Chapter is organised as follows: In section 7.2, the motivation of the space-time model is given, and the situations under which a space-time model may exhibit imporved performance over a time series model are described. In section 7.3, the concept of local spatio-temporal neighbourhood is introduced, and the method for incorporating it into the LOKRR model is described. The neighbourhood selection technique is outlined in section 7.4. In section 7.5, a case study is given of spatio-temporal forecasting of UTTs on the LCAP network, under the assumption of missing data. The aggregate results for the network are given in section 7.6, before three specific cases are examined in more detail in section 7.7. The Chapter is summarised in section 7.8.

7.2 Motivation

In Chapter 6, a local online kernel ridge regression model (LOKKR) was described for temporal forecasting of seasonal time series, which was applied to forecasting of travel times on urban road links. The model exhibits strong performance at short forecasting intervals, but its performance naturally diminishes as the forecasting interval increases. This is because it cannot forecast changes in traffic state that occur in the interval between the forecast being made and the forecast time instant. This situation also applies to times when data are missing, because forecasts must be made at increasing forecast horizons. This problem exists in all univariate models of traffic series and in univariate time series models in general. To avoid the necessity for longer range forecasts, additional information is required.

There are two sources of information that could be used to improve forecasts. Firstly, external data sources, such as incident data, weather data or other traffic variables from other sensors such as loop detectors or floating vehicles could be used. As a kernel based method, LOKRR is ideally suited to accommodate additional features of different types, but

this is not the focus of the current chapter[25]. The second option is to make use of observations of the series from different spatial locations in a spatio-temporal approach, with the aim of modelling the spatio-temporal evolution of traffic conditions across the network. There are many studies that make use of spatial information in order to improve traffic forecasts that are discussed in detail in chapter 4, which vary in their complexity (Ding et al., 2010; Kamarianakis and Prastacos, 2005; Kamarianakis et al., 2012; Min and Wynter, 2011; Min et al., 2010a, 2009b). The best performing approaches take into account the local properties and/or the dynamics of traffic as a network process. For instance, Kamarianakis et al. (2012), building on the work of Min and Wynter (2011) divide the data into distinct traffic regimes and estimate a sparse spatio-temporal neighbourhood for each regime. Cheng et al (2013) update the size and weighting of the spatio-temporal neighbourhood at each time step based on the prevailing traffic conditions.

The problem that these models address is that of identifying the *local or dynamic STN*, which is the the *neighbourhood of features or variables that is relevant to the location of interest at a given time*. If one can model the way in which the current process evolves from its past values, then one can model causation, which is the "holy grail of science" (Cressie and Wikle, 2011, p.297). In the purely temporal setting outlined in the previous chapter, the selection of features is relatively straightforward as the temporal dependence between observations tends to decrease with their temporal separation distance, or temporal lag. Therefore, a small subset of the immediate previous observations often provides enough information to produce a good forecast. However, in modelling a spatio-temporal process, the purely temporal approach is limited because it ignores spatio-temporal evolution. Temporal models are inherently reactive in that they can only model the effect of a sudden change in the series after the change has been observed. For instance, if an incident occurs on a given road link, its effects cannot be forecast until a spike in travel times is observed in the data. The effect of this worsens as the forecast horizon increases. In contrast, spatio-temporal models have the potential to enable proactive intervention. Like temporal models, they cannot forecast that an incident will occur, but if correctly specified, they can forecast the effects of the incident on neighbouring locations before these effects are observed in the data.

[25] Liu, (2008) provides an example of the use of signal timings and flow data to improve forecasting of travel times.

7.3 Local Spatio-temporal Neighbourhood

In most existing space-time models, the influence of the STN on the location of interest is considered to be static in space and time; that is they have a *globally fixed structure*. Furthermore, they usually make the assumption of *spatial isotropy*; that is that the influence from all directions is identical. However, in many space-time processes, it is not possible to achieve stationarity. In previous work (Cheng et al., 2011), it has been demonstrated empirically that the dependence between locations in road network data changes with time and across space. This motivates the incorporation into space-time models of STN structures that are *local* and/or dynamic in space and time (Kamarianakis et al., 2012; Cheng et al, 2013). A local STN can be incorporated into a space-time model through the construction of a time varying spatial weight matrix. First, consider a space-time series $z = \{z(s,t)|s \in S, t \in T\}$ in spatial domain S and temporal interval T collected at spatial locations $s = 1,2,\dots,S$ and times $t = 1,2,\dots,T$. In traditional space-time models such as STARIMA models, the spatial structure in S is incorporated into the model using one or more $S * S$ spatial weight matrices $\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^k$, where k is the maximum spatial order. Spatial weight matrices are usually binary, with $w_{i,j} = 1$ if locations i and j are spatially adjacent, 0 otherwise, although this need not be the case. To introduce temporal locality to the spatial weighting system, a time index is added to the spatial weight matrix, allowing its structure to vary with time: $\mathbf{W}_t^1, \mathbf{W}_t^2, \dots, \mathbf{W}_t^k$, where $t = 1,2,\dots,T$. For the purposes of this description, the superscript k will be dropped, but note that all explanations apply to matrices of higher orders.

The spatial neighbourhood of each of the S locations can change at each time instant. There are two ways in which this can happen. Firstly, elements that are zero at one time instant can become nonzero at a subsequent time instant. This equates to a change in the *adjacency structure* of the spatial weighting scheme. In the current context, this change may reflect a change in the direction of dependence from an upstream direction to a downstream direction. For example, when the network becomes congested downstream of a link, queues propagate in an upstream direction. The second way in which the spatial neighbourhood can change is that the value of the nonzero elements can change. This equates to a change in the *spatial weighting structure* of the spatio-temporal neighbourhood. This may happen if, for example, there is heavy congestion on one downstream link but not another. The congested downstream link will exert more influence on the travel times of the link in

question, and therefore will have a higher weight. In practice, the two situations outlined are more likely to be observed in the data where the spatial resolution is high.

In chapter 3, the distinction was made between spatio-temporal neighbourhood structures that are local and those that are dynamic. A local structure entails that the structure can change in time and across space, but that the overall structure is fixed based on historical conditions. This is the type of structure employed in, for example, Min and Wynter (2011) and Kamarianakis et al. (2012). A dynamic structure entails that the structure is both local and dynamic, meaning it can change based on the prevailing traffic condition. This type of structure is employed in, for example (Ding et al., 2010; Min et al., 2009b; Cheng et al, 2013). When building a space-time model of network data one must choose between these two structures based on their applicability to the network dataset under study. Those models that employ dynamic structures rely on assumptions about the physical relationship between observations at neighbouring locations.

The LCAP network is a spatially sparse network, where there are often long distances between sensor locations. Furthermore, as travel times are observed rather than flows, there is no concept of flow conservation. Therefore, it is more difficult to theorise a physical relationship between sensor locations that would enable dynamic STN updating[26]. For this reason, a local, rather than dynamic weighting structure is proposed here, in which the composition of the spatio-temporal neighbourhood is allowed to vary with time of day, but fixed in the training process.

7.4 STN selection

In chapter 5, the matrix $\mathbf{W}_{\mathrm{L,E}}$ was defined. $\mathbf{W}_{\mathrm{L,E}}$ is an $S * S$ distance matrix that provides an ordering of the network space and can be used to define the STN of the LCAP network. In the traffic forecasting literature, the STN is almost exclusively defined based on the position of sensors on a particular route. Therefore, there are clear definitions of upstream and downstream that can be used for STN selection. In this chapter, the case is considered where the immediate upstream and/or downstream information may not be available, or may be spatially distant. In this case, it is necessary to construct the spatio-temporal neighbourhood from the nearest available information.

[26] Note that the LSTARIMA model of Cheng et al. (2013) applied to the same network uses a dynamic structure, but this limits the STN to contain only those links connected in direct sequence in an upstream or downstream direction.

This can be done in a number of ways. Firstly, one can apply an inverse distance weighting to the entire weight matrix, and include all locations in a model. However, because of the spatio-temporal autocorrelation present in travel time data, it is not necessary to use all locations when forecasting a single link as the contribution of distant locations to a model is vanishingly small. The size of the spatial neighbourhood can be constrained using a distance threshold, the size of which needs to be determined. Alternatively, the *k*-NNs of a link can be used, in which case *k* needs to be determined. In the context of network data with uneven link lengths, setting a distance threshold is problematic since the distances between locations vary across space. In the city centre where the network is dense, a link may have many neighbours within 500m, but on the periphery a link may have no neighbours within 5 kilometres. Therefore, k-nearest neighbours is used as the spatial neighbourhood selection technique in this case.

7.4.1 K-nearest neighbours STN selection

The *k*-nearest neighbours of each spatial location are selected from the distance matrix $\mathbf{W}_{\mathrm{L,E}}$. To simplify this procedure, a new *S*S* matrix is constructed called $\mathbf{NN}_{\mathrm{L,E}}$. The i, j element of $\mathbf{NN}_{\mathrm{L,E}}$ is the ID of the jth nearest neighbor of link i. Therefore, the first column of $\mathbf{NN}_{\mathrm{L,E}}$ contains the IDs of all the first order nearest neighbours of each of the links, the second column contains the IDs of the second order nearest neighbours, and so on. To select the k-NNs of all the road links in the network, one simply selects the first *k* columns of $\mathbf{NN}_{\mathrm{L,E}}$. The previous *m* observations of the k-NNs of each link are then used to create the pattern set:

$$\{(\mathbf{x}_{11}, y_{11}), (\mathbf{x}_{12}, y_{12}), \dots, (\mathbf{x}_{1t}, y_{1t}), (\mathbf{x}_{21}, y_{21}) \dots, (\mathbf{x}_{st}, y_{st})\}, \tag{7.1}$$

With:

$$\mathbf{x}_{st} = \left[\{y_t^s, y_{t-1}^s, \dots, y_{t-(m-1)}^s\}, \{y_t^{s+1}, y_{t-1}^{s+1}, \dots, y_{t-(m-1)}^{s+1}\}, \dots, \{y_t^{s+(k-1)}, y_{t-1}^{s+(k-1)}, \dots, y_{t-(m-1)}^{s+(k-1)}\}, \hat{\mu}_t^s\right] \tag{7.2}$$

Where *s* is the spatial index, *t* is the time index, *m* is the embedding dimension, *k* is the number of neighbours and $\hat{\mu}_t^s$ is the sample mean calculated from the historical data according to equation 6.30. Equation 7.2 concatenates the previous *m* observations at

location *s* with the observations at its *k* nearest neighbours. Note that, although nearest neighbours are used in this specification, there are various other ways in which the relevant neighbours could be selected, which are discussed in the following Chapter.

Under missing data, the observations of location *s* are assumed to be unavailable. Therefore, $\mathbf{x}_{st}$ becomes:

$$\mathbf{x}_{st} = \left[\{y_t^{s+1}, y_{t-1}^{s+1}, \dots, y_{t-(m-1)}^{s+1}\}, \{y_t^{s+2}, y_{t-1}^{s+2}, \dots, y_{t-(m-1)}^{s+2}\}, \dots, \{y_t^{s+(k-1)}, y_{t-1}^{s+(k-1)}, \dots, y_{t-(m-1)}^{s+(k-1)}\}, \hat{\mu}_t^s\right] \quad (7.3)$$

In equation 7.3, the future values of location *s* are modelled as a combination of previous values recorded at *k* neighbouring locations. When data are missing at location *s*, only the independent variables of the data pattern $(\mathbf{x}_{st}, y_{st})$ are observed. Therefore, the dependent variable is replaced with the *corresponding forecast value*.

7.5 Case Study

In this section, the space-time (ST)LOKRR model is applied to forecasting of travel times on the LCAP network under three missing data scenarios. In section 7.5.1, the missing data scenarios are described. Following this, a data description is provided in section 7.5.2. The model training procedure is described in section 7.5.3. Finally, in section 7.5.4, the indices used for the evaluation of model performance are described.

7.5.1 Missing data scenarios

The aim of this experiment is to test the performance of the STLOKRR model in forecasting under missing data caused by sensor failure, which has been shown to be a serious problem on the LCAP network in Chapter 4. Each LCAP link connects two cameras. Both of these cameras must be functioning in order for observations to be made. There are three situations in which sensor failure would prevent observations: 1) the camera at the beginning of the link has failed; 2) the camera at the end of the link has failed; or, 3) both cameras have failed. The latter situation is the most serious as multiple links are often served by the same camera, which means that, if a single camera fails, there is often information collected on another link that is partially shared with the link in question. However, if both cameras on a link fail then those other links that use the same cameras will also fail to collect information and will not be useable in forecasting. This can lead to a large

spatial gap in data collection. All three scenarios are tested in order to assess the robustness of the approach to multiple sensor failures, and will henceforth be referred to as scenario 1, scenario 2 and scenario 3.

7.5.2 Data Description

Testing of the spatio-temporal model is focused on those LCAP links that lie within London's Congestion Charging Zone (CCZ). There are two reasons for this, firstly, the spatial coverage of the network is most dense within the CCZ, and secondly, the quality of the data is high within the CCZ due to the priority of TfL to monitor vehicles for Congestion Charging purposes. To ensure the consistency of the results of the analysis, only those links that are considered to be "core" links by TfL are forecast, of which there are 80 in the CCZ (in the cleaned dataset, see section 4.3.6). These are links that are considered to be strategically important. Non-core links are used in forecasting if they fall within the spatio-temporal neighbourhood of a core link. The network of core links within the CCZ is shown in figure 7.1, overlaid on the complete LCAP network.

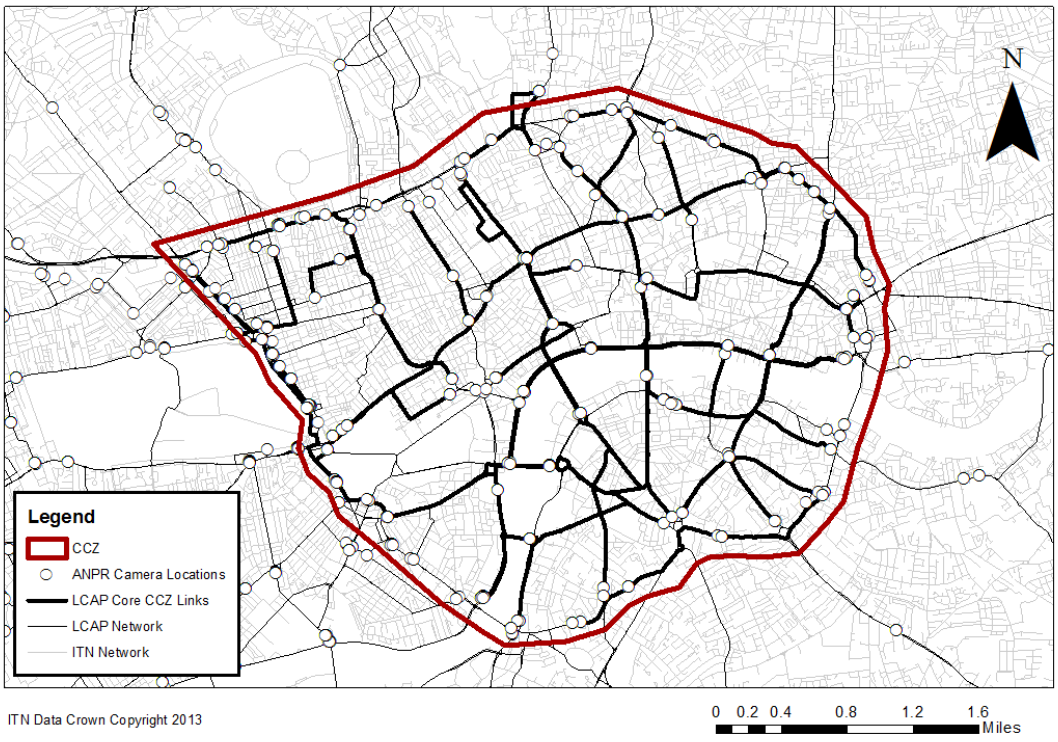


Figure 7.1 – Network of core LCAP links within the CCZ

Table 7.1 gives some summary statistics about the test network. The overall missing rate is 13.7%, but some links display patch rates of up to 42%. Given that this is the missing rate for

core links that are considered to be strategically important, missing data clearly warrants treatment.

Table 7.1 – Summary statistics of the test network

Avg Length	s.d. length	min length	max length	Mean Patch	PT 1 %	PT 2 %	PT 3 %
1228.531	581.3895	226.1372	3384.127	0.24	5.6	5.4	2.7

PT = Patch type

To maintain consistency with the analysis in Chapter 6, the data are divided into three sets. The training data is 80 days and the testing data is 37 days, between Tuesday 28th April and Wednesday 1st June 2011. Finally, another 37 days are used to validate the model performance, between Thursday 2nd June and Friday 8th July 2011. The best performing model on the testing set in terms of the RMSE index is used to forecast the validation set. The performance on the testing set is referred to as training performance and the performance on the validation set is referred to as testing performance.

7.5.3 Model training

To train the models, the parameter space that was used in chapter 6 is used for the temporal parameters: $\lambda = \{\frac{1}{8}\lambda_0, \frac{1}{4}\lambda_0, \frac{1}{2}\lambda_0, \lambda_0, 2\lambda_0\}$, $\sigma = \{q1, q2, q3\}$, which are the 0.25, 0.5 and 0.75 quantiles of $\|\mathbf{x} - \mathbf{x}'\|^2$, and $w = \{1,2,3\}$. Again, m is fixed at 3. In the spatio-temporal model there is an additional parameter to train, *k*, which is the size of the spatio-temporal neighbourhood. Errors are calculated both for a ***global model***, where a single set of parameters are determined for the model, and a ***local model***, where the parameters are allowed to vary by time of day. The results for each of the models are determined during the same training process. The performance of these two models is compared in order to assess the effect of the time varying parameters.

7.5.4 Performance evaluation

The NRMSE and MAPE are used to measure the performance of the models, along with an additional index called $MASE_{hist}$. In chapter 6, the mean absolute scaled error (MASE) was described as a generally applicable measure of forecast accuracy (Hyndman and Koehler, 2006). The MASE compares the absolute errors in the forecast values with the absolute errors of a naïve forecast, where the previous value of the series is used to forecast the next

value: $\hat{y}_t = y_{t-1}$. This index is applicable for short term forecasts, where a recent observation is available. However, for phenomena such as urban travel times that exhibit a cyclical pattern it does not make sense to use such a forecast when the forecast horizon is large. This is because the historical average of the series would be expected to provide a better forecast. To address this, a modified version of the MASE is defined here, called $MASE_{hist}$ that compares the performance of the forecast model to the naïve historical average predictor:

$$MASE_{hist} = 1/n \sum_{t=1}^{n} \left(\frac{|e_t|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{hist}|} \right) \tag{7.1}$$

Where e_t is the forecast error for a given period and the denominator is the average forecast error of the naïve historical average method. Again, a $MASE_{hist}$ value less than 1 indicates that the forecast model outperforms the naïve model, while a value greater than 1 indicates the opposite. To be viable, any model of missing data should outperform the historical average model as a minimum requirement.

7.5.5 Benchmark Model

To further assess the viability of the model, its performance is compared with a benchmark model, *k*-NN. k-NN is among the simplest of all machine learning algorithms, and has been widely used in classification and regression (Mitchell, 1997). Despite its simplicity, it has shown strong performance in forecasting of traffic series in the past (Smith and Demetsky, 1996, 1997). *k*-NN has been used in Haworth and Cheng (2012) along with kernel regression for forecasting under missing data on the LCAP network. Only *k*-NN is used here as its performance has been found to be very similar to kernel regression. As a pattern based algorithm, *k*-NN is similar in nature to LOKRR, and provides a good benchmark for the performance of pattern matching techniques in the current setting. The basic idea is to compute the distance (usually Euclidean) between a test sample and each of the samples in a training set:

$$d_i = \|\mathbf{x}_t - \mathbf{x}_i\| \tag{7.2}$$

Where $i = 1,2, \dots, D$ are the training data. A forecast is made as a function of the k nearest samples in the training set. This function can take many forms, the simplest of which is to produce the output as the arithmetic mean of the neighbours:

$$y_{t+1} = 1/k \sum_{i=1}^{k} y_i \tag{7.3}$$

Where K is the number of neighbours. To train the *k*-NN model, *k* is varied between 1 and 20 as performance increase was found to plateau at 20 in Haworth and Cheng (2012). The number of spatial neighbours is varied between 1 and 8 to maintain consistency with the LOKRR results. To avoid confusion, the number of neighbours in the *k*-NN algorithm is denoted k_n and the number of spatial neighbours is denoted k_s. The model is implemented using the *caret* library in R statistical package, which provides an interface to various forecasting algorithms (Kuhn, 2008).

7.6 Results

In this section, the experimental results are presented. The full link by link results of the training and testing phases for individual links with global and local parameters are given in Appendix B. The average results are summarised in tables 7.2 and 7.3. The average performance of the global model is fairly consistent across the three scenarios. As expected, the performance under scenario 3 is poorer than the performance under scenarios 1 and 2, but the difference is slight. This indicates that, although there is less information coming from the immediate spatio-temporal neighbourhood of each link in scenario 3, there is still enough relevant information available to provide a viable forecast in most cases. During the training periods, the percentage of links for which the MASEAvg is less than 1 is 90, 88.8 and 86.2 for scenarios 1, 2 and 3 respectively. This falls to 65, 70 and 58.8 in the testing period. Although the MASEAvg is below 1 for the majority of links in all cases, the decrease in performance between the training and testing periods indicates that some overfitting may have taken place. However, examining the average NRMSE values, it can be seen that better performance was achieved in the testing phase than in the training phase for all scenarios. This indicates that the traffic conditions in the testing phase were more forecastable than those in the training phase. The results in terms of MAPE remain stable at around 20-22% for all links in both the training and testing phases.

Table 7.2 – Summary of training performance

Scenario	Global				Local			
	NRMSE	MAPE	MASEAvg	MASEAvg %	NRMSE	MAPE	MASEAvg	MASEAvg %
1	0.0802	21.16	0.8914	90	0.07691755	20.28616	0.792515	95
2	0.0799	20.84	0.8874	88.8	0.07711880	20.18834	0.793425	96.3
3	0.0835	21.79	0.9215	86.2	0.08079056	21.05833	0.823907	93.8

Table 7.3 – Summary of testing performance

Scenario	Global				Local			
	NRMSE	MAPE	MASEAvg	MASEAvg %	NRMSE	MAPE	MASEAvg	MASEAvg %
1	0.0727	21.03	0.9396	65	0.081299	21.70701	1.06382	15
2	0.0722	21.01	0.9404	70	0.08233	22.11098	1.075838	15
3	0.0751	21.99	0.9757	58.8	0.085133	22.96141	1.107536	7.5

The training performance of the local model is superior to its testing performance, and the global model produces lower average testing errors than the local model across all indices. This result indicates that the local model overfits the training data. There are several factors that contribute to this result. Firstly, in the global parametric setting, there are approximately 37*180=6660 training points available to fit the model, whereas in the local setting there are just 37 points for each time of day. This means that there is a strong possibility of extreme values influencing the chosen model. Secondly, the sparseness of the spatial structure means that the dynamics of traffic are not fully observed. Therefore, changes in the size of the spatial neighbourhood are less likely to be observed in the data in a deterministic manner. The MASEAvg % for the local model is 15, 15 and 7.5 in scenarios 1, 2 and 3 respectively, demonstrating that the historical average provides a better forecast than the locally trained model in the majority of cases. It can be concluded, therefore, that the local model is not a viable forecasting model on this dataset, with this training scheme. However, it should be noted that the local model outperforms the global model on 12, 12 and 6 of the 80 links in scenarios 1, 2 and 3 respectively.

7.6.1 Global Parameters

Tables 7.4 - 7.6 summarise the global fitted model parameters for each of the scenarios. The number of spatial neighbours *k* varies between 1 and 8. However, 77.5% of the models across all scenarios exhibit best performance at $k \leq 5$. Only 11 of the 240 models exhibit best performance at *k*=8, suggesting that the maximum value of *k* is sufficient to capture the relevant spatio-temporal information. The trained values of *w* vary between 1 and 3, but the majority (80%) of the best models are found with w=3. This result is similar to that obtained in the temporal forecasting experiment in chapter 6 at longer forecast horizons. The implication is that, with less knowledge of the local conditions on the link, more information is required about the historical conditions in the spatial neighbourhood. In terms of the kernel parameter σ, the majority of best performing models are found at σ = 1 (68.3%), followed by σ = 3 (20%). Those models that exhibit best performance at σ = 1 make more use of the local spatio-temporal information because a smaller value of σ indicates a more complex solution. Those models with σ = 3 produce forecasts that are closer to the historical average. The fitted value of λ was found to be $1/8\lambda_0$ in all cases, mirroring the result of the temporal model in Chapter 6.

Table 7.4 – Fitted values of neighbourhood size k for each scenario. The numbers in the cells are the count of best performing models for each value of k.

	Scenario		
k	**1**	**2**	**3**
1	17	18	16
2	13	18	22
3	17	24	15
4	10	5	11
5	8	4	10
6	5	5	1
7	4	4	2
8	6	2	3

Table 7.5 – Fitted values of w for each scenario. The numbers in the cells are the count of best performing models for each value of w.

	Scenario		
w	**1**	**2**	**3**
1	5	4	7
2	7	9	15
3	68	67	58

Table 7.6 – Fitted values of σ for each scenario. The numbers in the cells are the count of best performing models for each value of σ.

	Scenario		
σ	**1**	**2**	**3**
1	54	55	55
2	8	11	9
3	18	14	16

7.6.2 Comparison with the benchmark model

Tables 7.7 and 7.8 show the average training and testing errors of the k-NN method respectively. In the training period, k-NN performs well and outperforms the historical average method in nearly all cases across the three scenarios, mirroring the performance of STLOKRR. The performance drops significantly in the testing phase, where the MASEAvg is less than 1 for 63.75%, 65% and 51.25% of the links in scenarios 1, 2 and 3 respectively. This result, again, mirrors that of the STLOKRR model. The overall MASEAvg for scenario 3 is greater than 1, indicating that the historical average is a better model.

In the testing phase, STLOKRR outperforms the k-NN method in terms of NRMSE in 92.5%, 87.5% and 91.25% of cases in scenarios 1, 2 and 3 respectively. This indicates that STLOKRR has better generalisation ability than k-NN.

Table 7.7 – Training errors for the k-NN model in each of the scenarios

Scenario	NRMSE	MAPE	MASEAvg	MASEAvg %
1	0.076424	20.99677	0.813198	97.5
2	0.076205	20.89881	0.80864	97.5
3	0.078844	21.6293	0.833246	96.25

Table 7.8 – Training errors for the k-NN model in each of the scenarios

Scenario	NRMSE	MAPE	MASEAvg	MASEAvg %
1	0.078421	23.38032	0.971459	63.75
2	0.078212	23.32616	0.972597	65
3	0.082375	24.55182	1.015005	51.25

7.7 Examples of model performance

In this section, three examples are shown of the performance of the model at forecasting under missing data. As the global model significantly outperforms the local model, only the global results are presented here. The section is split into three subsections, each of which covers a single case study. In each subsection, a detailed description is first given of the case study area. Following this, the predictive results are presented. The case studies are chosen to illustrate the range of forecasting performance of the model. The reasons for the variation in performance are explored.

7.7.1 Example 1: LCAP Link 417 - A10 Bishop's Gate, North East Bound

The first example is the A10, Bishop's Gate, North East bound, which is located north of the River Thames in the eastern part of the CCZ, shown in figure 7.2. The LCAP link marked in red is the link to be forecast (LCAP 417). The links that are connected to the start camera of link 417 are:

433, 434, 446, 1407, 1604, 2357, 2358

The links that are connected to the end camera of link 417 are:

556, 564, 1436, 1441, 1604, 2446

Note that link 1604 shares both the start and end camera of link 417 as it runs in parallel in the opposite direction. For each of the three scenarios, the best model was found with k=3. Examining the results in detail, it transpires that the best global model for each scenario is identical, consisting of links 447 (green line), 2043 (orange line) and 455 (blue line), listed in order of distance from the midpoint of link 417.

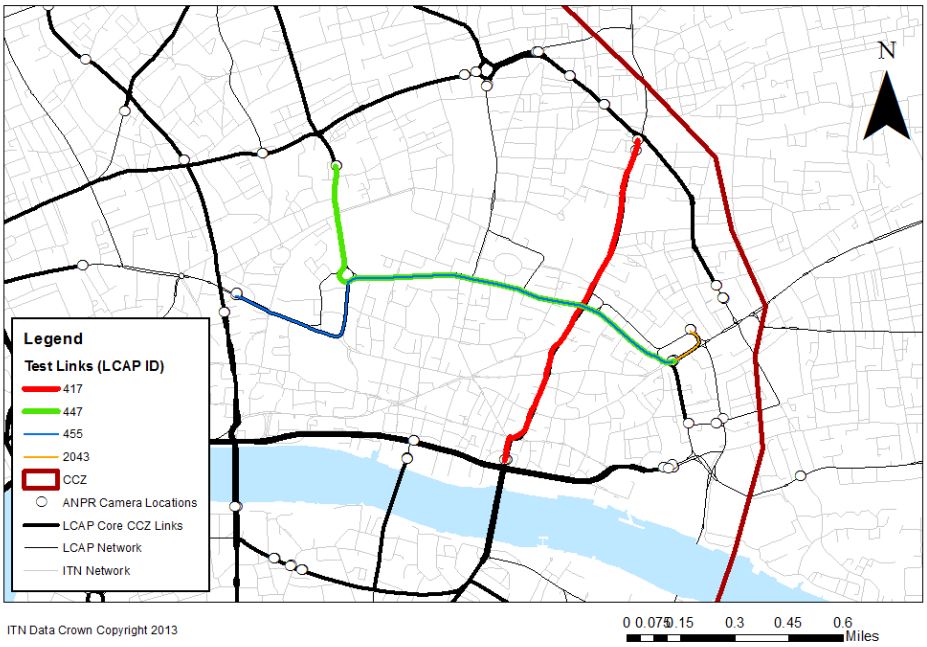


Figure 7.2 – Location of link 417 and its spatial neighbours

Table 7.9 gives details of link 417, and the three neighbours in the trained model. Links 447 and 455 partially overlap as they share the same start camera. It is interesting to note that the links directly connected in the upstream and downstream directions are not selected in the best performing models in scenarios 1 and 2, where they are available. From the perspective of traffic flow theory, one would expect links connected in sequence to have a strong influence on one another. This type of spatio-temporal relationship is not revealed in the data here because of the sparsity of the sensor network and the length of the links. Link 417 is 1614.57 metres in length and contains several signalised intersections. This means that conditions on links beyond its start and end points are spatially distant, particularly if those links are also long. In contrast, links 447 and 445 interact directly with the traffic flow at the centre of link 417. This highlights the importance of the neighbourhood specification used here.

Table 7.9 – Details of neighbours of link 417

LCAP ID	Start Camera	End Camera	Length (m)	Core	Name	Direction
417	186tfl	019tfl	1614.57	Yes	A10 Bishops Gate NEB	NEB
2043	007tfl	006tfl	226.1372	Yes	A1211 St Botolph St / Aldgate High St SWB	SWB
447	006tfl	187tfl	2118.686	No	A1211 London Wall / A1 Aldersgate St NWB	NWB
455	006tfl	192tfl	2307.456	Yes	A1211 London Wall NWB	NWB

Links 447 and 455 intersect with link 417 near to its midpoint. The intersection is shown in figure 7.3. The image is captured from link 417 looking in the Northbound direction. Turns are allowed in both directions, except when approaching from the South, where there is no right turn. One observation that can be made is that there is no hatched area on the intersection meaning that it is possible that vehicles may occupy the intersection between signal cycles during congestion. This means that heavy traffic on one route may lead to a build-up of traffic on the other route. Furthermore, traffic from links 447 and 455 can join link 417, thus directly influencing its level of flow.

Table 7.10 – Training results, link 417

Scenario	k	w	sigma	RMSE	NRMSE	MAPE	MASEHist
1	3	3	1	0.041	0.093	14.83	0.96
2	3	3	1	0.041	0.093	14.83	0.96
3	3	3	1	0.041	0.093	14.83	0.96

Table 7.11 – Testing results, link 417

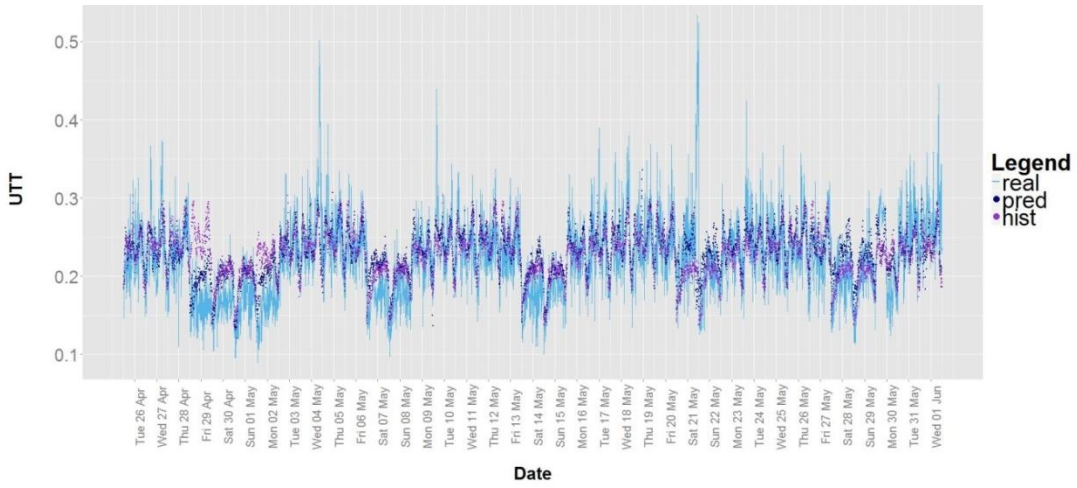
Scenario	RMSE	NRMSE	MAPE	MASEHist
1	0.044	0.071	12.56	0.98
2	0.044	0.071	12.56	0.98
3	0.044	0.071	12.56	0.98

Figure 7.3 – Google street view snapshot of the intersection of links 417 and 447/455 (taken 2008).

Tables 7.10 and 7.11 show the training and testing results for each of the scenarios. As the trained model was identical in each case, the results are the same for each scenario. Consistent with the global results presented in section 7.6, the performance in the testing phase is superior to that in the training phase in terms of NRMSE and MAPE. This coincides with a slight decrease in performance in terms of MASEHist, indicating that the conditions in the testing phase are more forecastable than those in the training phase. The values of the other model parameters were w=3, σ=1 and λ=1/8.

Figure 7.4 shows the real versus fitted values for link 417 for the training period. Despite there being no local temporal information available pertaining to the traffic conditions on link 417, the forecast values match the observed values well in general. The 4th and 7th days of the training period, which are a Friday May 29th and Monday June 2nd, were the Royal Wedding bank holiday and the early May bank holiday respectively (public holidays). As one would expect, abnormally low travel times were observed on these days. The model is able account for this, although the evening peak on the Friday is slightly over estimated. Overall, the model is able to forecast the daily variation in travel times well. However, it is not able to forecast the large peak travel times that occur on some of the days, such as on the 26th day. The reason for this is that the peak exhibited on that day was spatially isolated on link 417 and was not forecastable from the data. For other abnormal peaks, such as on the 35th day, the model performs better.

a)



b)

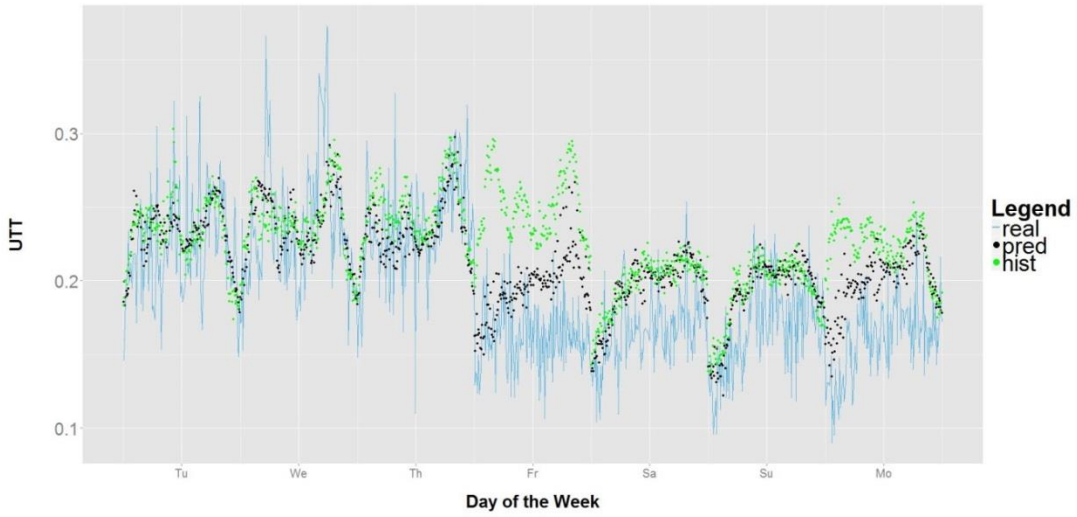
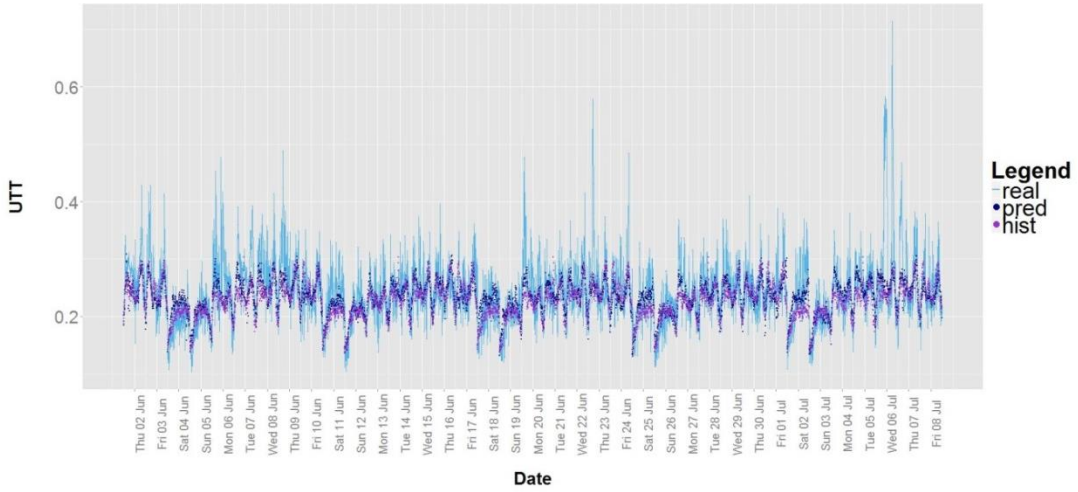


Figure 7.4 –Real versus fitted values for link 417, a) across the entirety of the training period and; b) in the first week of the training period

a)



b)

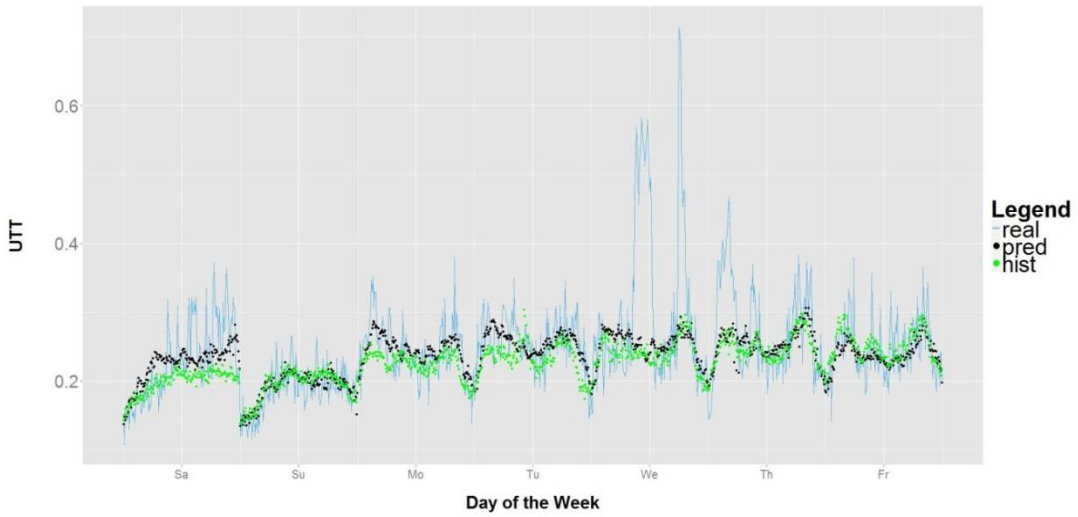


Figure 7.5 –Real versus forecast values for link 417, a) across the entirety of the forecast period and; b) in the final week of the testing period

Figure 7.5 shows the testing performance, which is broadly similar to the training performance. Figure 7.5 b) shows the real versus forecast values in the final week of the testing period. It can be seen that, although the model is able to forecast traffic conditions that are close to the historical average fairly accurately, it is not able to forecast the peaks that occurred on the Wednesday and Thursday of this week. In general, the model produces forecasts that are close to the historical average on link 417. The main reason for this is that the observed travel times are very consistent from day to day on this link. Therefore, a forecast close to the historical average is usually a good forecast. However, the model is still viable since the MASEAvg is less than 1.

7.7.2 Case Study 2: LCAP Link 1745 – A3211 Victoria Embankment, North East Bound

The second case study involves LCAP link 1745, which is the A3211 Victoria Embankment, north east bound. There are 15 links that are connected to the start camera of link 1745, excluding itself:

1613, 2344, 439, 424, 2778, 2780, 2847, 1747, 1622, 2318, 475, 476, 2777, 2781, 2846

There are just 4 links that are connected to the end camera of link 1745:

1453, 1745, 2848, 2854

Tables 7.12 and 7.13 show the performance of the model in the training and testing phases. The performance is far poorer in scenarios 1 and 3 than in scenario 2, with MASEAvg values of 1.24 for both scenarios 1 and 3, compared with 0.9 for scenario 2. The real versus forecast values for the training phase are shown in figures 7.6 a)-c). It can be seen that link 1745 displays a high degree of variability in the size of its peak travel times. In scenarios 1 and 3, the model frequently over forecasts the peak period from the 19th of May onwards. This is not the case in scenario 2.

Table 7.12 - Training results, link 1745

Scenario	k	W	sigma	RMSE	NRMSE	MAPE	MASEHist
1	8	2	3	0.094716	0.110708	39.61017	1.240681
2	2	2	2	0.0797	0.0932	24.5497	0.9038
3	8	2	3	0.094639	0.110617	39.53508	1.239572

Table 7.13 – Testing results, link 1745

Scenario	RMSE	NRMSE	MAPE	MASEHist
1	0.1145	0.0837	29.98	1.0653
2	0.1069	0.0781	25.77	0.9610
3	0.1145	0.0838	29.98	1.0654

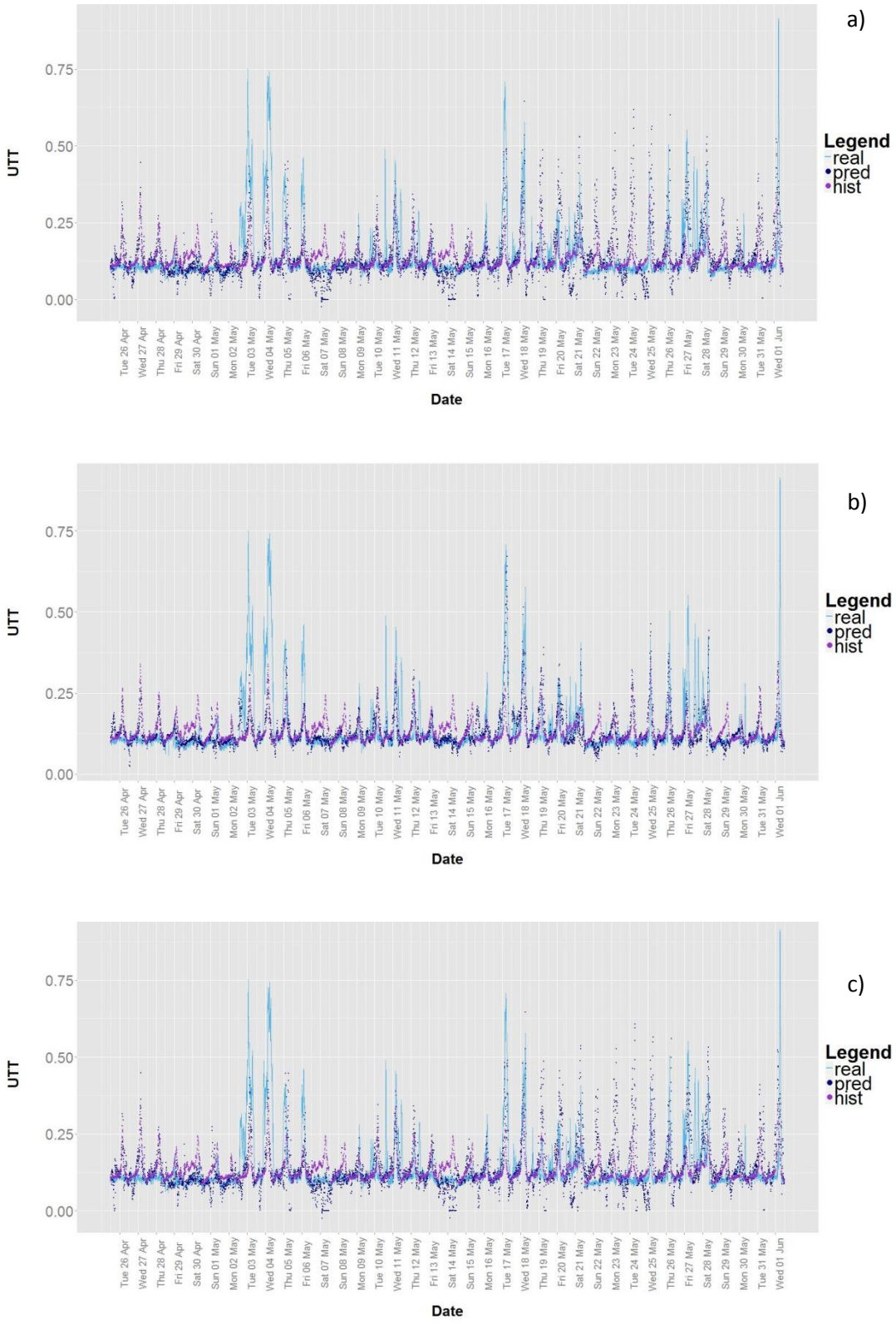


Figure 7.6 – Real versus fitted values for link 1745 for; a) scenario 1; b) scenario 2; and c) scenario 3.

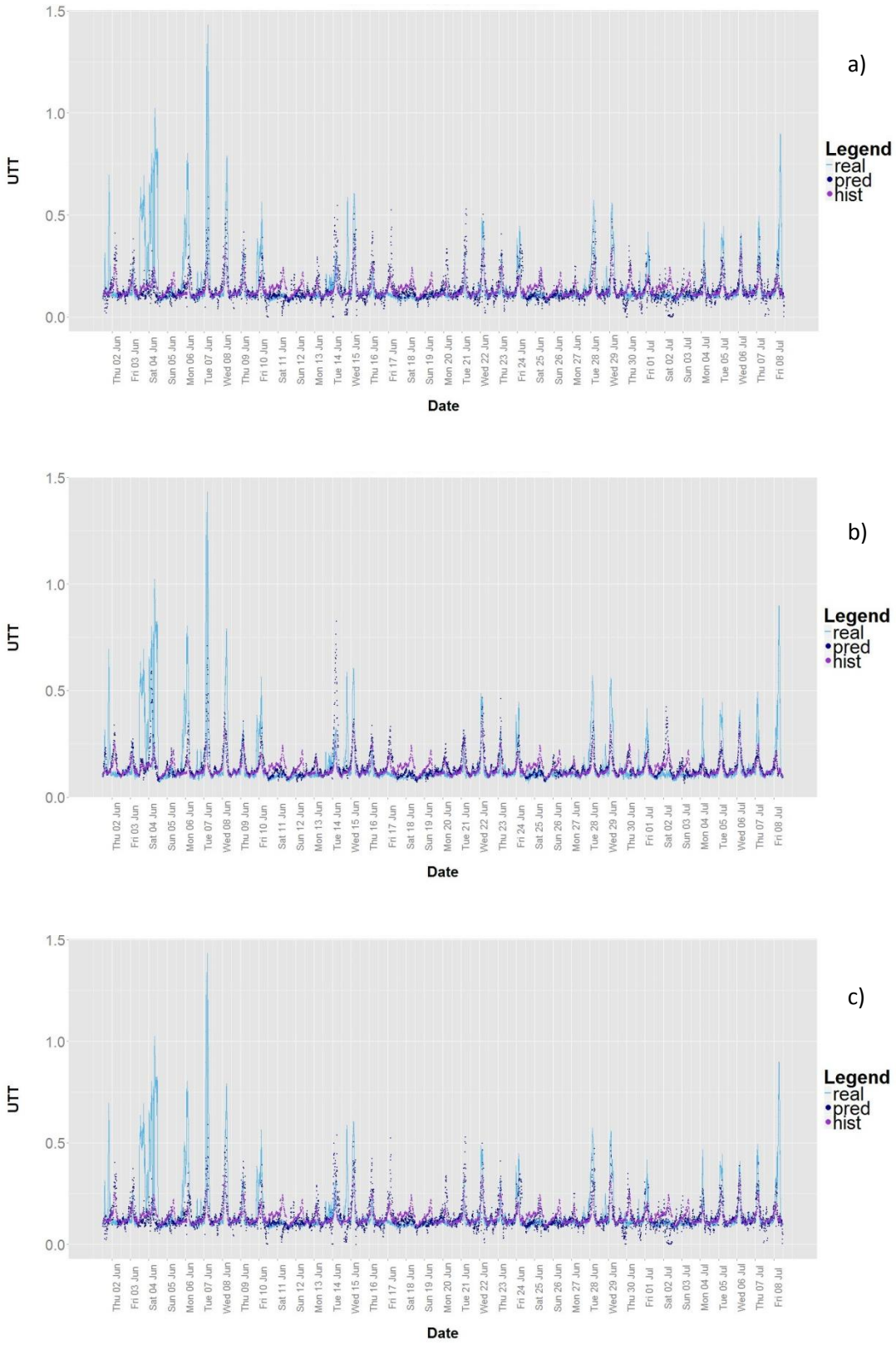


Figure 7.7 –Real versus forecast values for link 1745 for; a) scenario 1; b) scenario 2; and c) scenario 3.

Figures 7.7 a)-c) show the real versus forecast values for each of the scenarios in the testing phase. In general, the performance in the testing phase is superior to that in the training phase. There are fewer large over forecasts of the peak travel times that were seen in the training period in scenarios 1 and 2. The models are able to forecast some of the peaks accurately, for instance on the 7th and 8th of June. However, the sizes of other peaks are overestimated, such as Tuesday 14th June, particularly under scenario 2. This result implies that the spatio-temporal relationship between link 1745 and its neighbours is not fully captured in the data. The presence or absence of congestion in the neighbourhood of link 1745 is not a consistent predictor of congestion on link 1745.

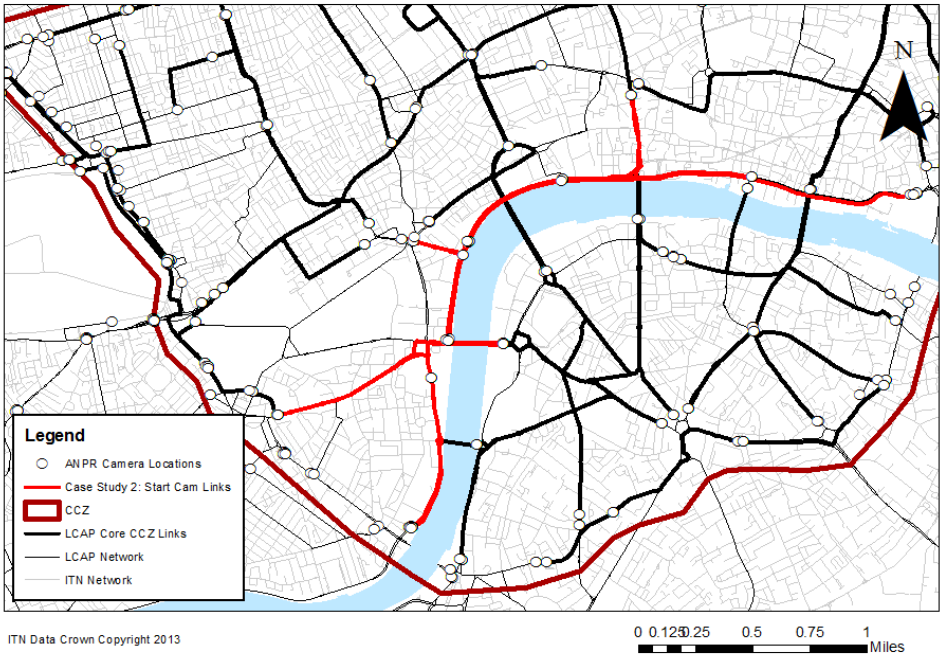


Figure 7.8 – Links connected to the start camera of LCAP link 1745

The links connected to the start camera of link 1745 (203tfl) are shown in red in figure 7.8. Three of the 15 links are core links. The area of London that is monitored by links connected to this camera is quite large, extending along the north bank of the Thames from just north of Vauxhall Bridge to just west of Tower Bridge, a distance of approximately 3.5 miles. Therefore, the implications of camera failure at this location are severe. This is reflected in the predictive results when camera 203tfl is removed. The value of k for scenarios 1 and 3 is 8, whereas it is just 2 for scenario 2. The two best predictors of link 1745 are connected to tfl203. The neighbours for each scenario are shown in figure 7.9 a) and b). They are not available in scenarios 1 and 3. Because of this, camera 203tfl can be seen as a critical sensor in the network. Table 7.14 provides details of the links in question.

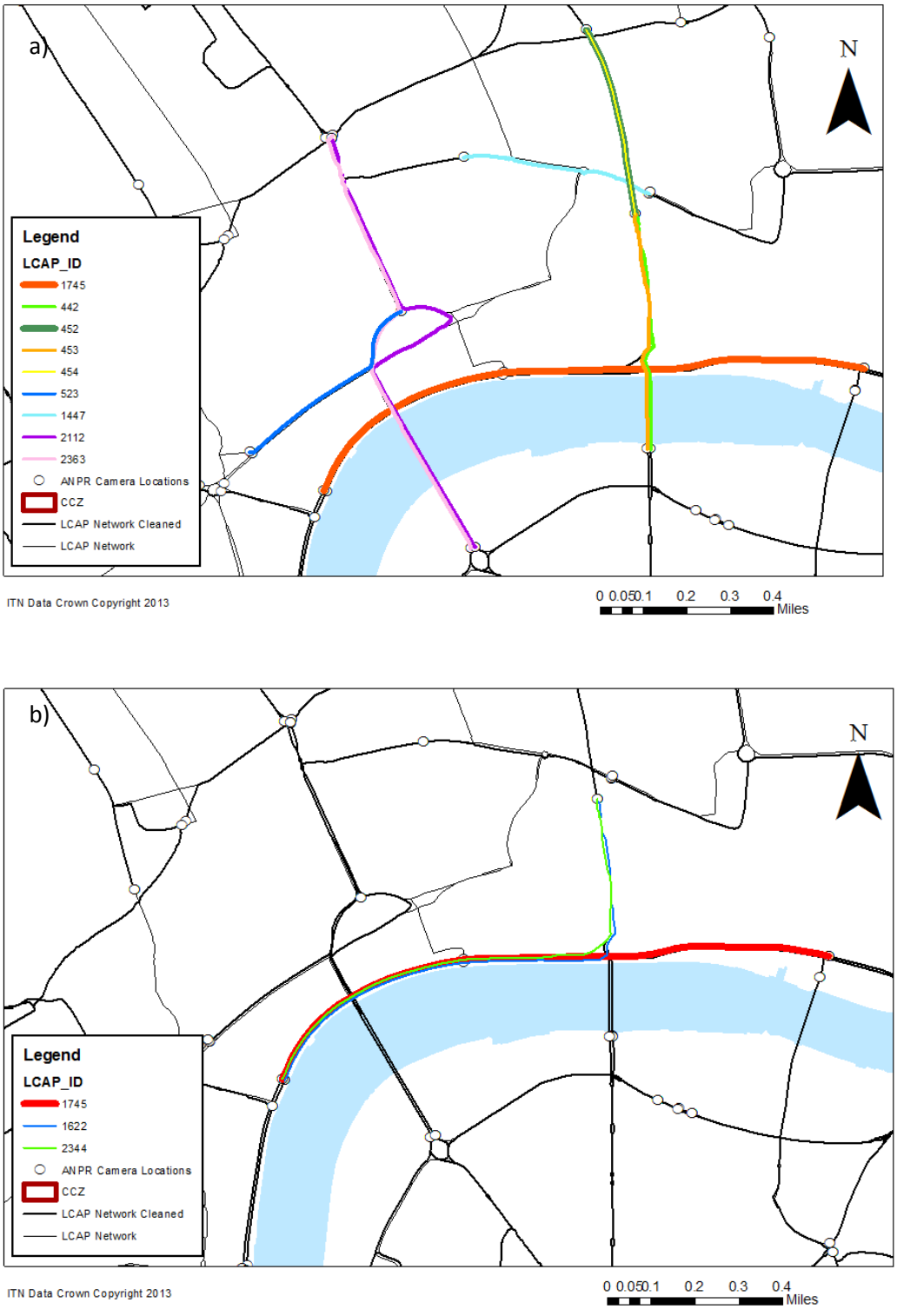


Figure 7.9 – Selected neighbours for link 1745: a) scenarios 1 and 3 (k=8); b) scenario 2 (k=2)

The importance of camera 203tfl in the network is not the sole reason for the poor forecasting performance on link 1745. The end camera of link 1745 is connected to only three other links, one of which is a core link. It can be seen from figures 7.6 b) and 7.7 b)

that, although some of the peak travel times are forecast accurately, other peaks are significantly underestimated, for instance on the 8th, 9th, 10th and 11th days of the training period.

Table 7.14 – Details of neighbours of Link 1745

LCAP ID	Start Camera	End Camera	Length (m)	Core	Name	Direction	Scenarios
1745	203tfl	205tfl	2185	Yes	A3211 Victoria Embankment / Upper Thames St NEB	East	1,3
1447	192tfl	521tfl	729	No	A40 High Holborn NWB	N/A	1,3
2112	202tfl	183tfl	1987	Yes	A4200 Kingsway / A301 Waterloo Bridge Rd SEB	N/A	1,3
2363	183tfl	202tfl	1729	Yes	A301 Waterloo Bridge / Kingsway NWB	N/A	1,3
523	198tfl	522tfl	803	Yes	A4 Strand NEB	N/A	1,3
442	191tfl	184tfl	899	Yes	A201 Farringdon / A201 Blackfriars Bridge SEB	South	1,3
452	191tfl	189tfl	710	No	A201 Farringdon Rd NWB	N/A	1,3
453	184tfl	191tfl	893	Yes	A201 Blackfriars Bridge / Farringdon Rd NWB	North	1,3
454	189tfl	191tfl	710	Yes	A201 Farringdon Rd SEB	South	1,3
1745	203tfl	205tfl	2185	Yes	A3211 Victoria Embankment / Upper Thames St NEB	East	2
1622	191tfl	203tfl	1956	No	A201 Farringdon Rd / A3211 Victoria Embankment SWB	N/A	2
2344	203tfl	191tfl	1912	No	A3211 Embankment / Farringdon Rd NEB	N/A	2

Examining the physical characteristics of link 1745, it is restricted to the South by the River Thames, and passes under the two road bridges that occur along its length. This means that flows into and out of it are constrained by its physical location. Therefore, much of the congestion that occurs on link 1745 begins on link 1745 and doesn't spread spatially, making its values inherently unpredictable without local temporal information. However, the MASEAvg for both the training and testing periods for scenario 2 is less than one, showing an improvement over the naïve historical average. This means that the model for scenario 2 is a viable model.

7.7.3 Case Study 3: Link 1747 - A3211 Tower Hill / A3211 Embankment, South West Bound

The third case study involves link 1747, which is A3211 Tower Hill / A3211 Embankment South West Bound. Link 1747 partially overlaps with link 1745 (case study 2) but flows in the opposite direction. There are 15 links that are connected to the start camera of link 1747, excluding itself:

1745, 1613, 2344, 439, 424, 2778, 2780, 2847, 1622, 2318, 475, 476, 2777, 2781, 2846

There are 16 links that are connected to the end camera of link 1745:

1878, 2398, 444, 2513, 482, 484, 2849, 1877, 1618, 1619, 1620, 1026, 1453, 2397, 2547, 2512

In contrast to link 1745, link 1747 displays strong forecasting performance. Tables 7.15 and 7.16 show the training and testing errors, along with the fitted model parameters. Each model has a different value of *k*. The selected neighbours are shown in table 7.17.

Table 7.15 –Training results, link 1747

Scenario	k	W	sigma	RMSE	NRMSE	MAPE	MASEHist
1	5	3	1	0.075	0.109	20.70	0.80
2	8	3	1	0.064	0.093	19.92	0.72
3	4	3	1	0.073	0.106	20.39	0.79

Table 7.16 –Testing results, link 1747

Scenario	RMSE	NRMSE	MAPE	MASEHist
1	0.067	0.095	23.23	0.9547
2	0.063	0.088	21.97	0.9157
3	0.065	0.092	23.12	0.9453

Interestingly, the model that exhibits the best performance in scenario 2 (and the best performance overall) has *k*=8. The selected neighbours are shown in figure 7.10 b). The 6^{th}, 7^{th} and 8^{th} nearest neighbours are links 1620, 1453 and 444 respectively. These neighbours share the same section of road as link 1747, which explains the improvement in performance that their inclusion brings. It also highlights a drawback in the distance measure, because links whose midpoints are closer to the midpoint of link 1747 are

considered to be the closest neighbours, despite not sharing the same route. However, in the case study presented in section 7.7.1 involving link 417, links that were not on the same route were found to produce the best performance. Additionally, links 1620, 1453 and 444 all share the start camera of link 1747 and hence are not available for forecasting in scenarios 1 and 3, and good performance is also achieved in these scenarios.

Table 7.17 - Details of neighbours of Link 1745

LCAP ID	Start Camera	End Camera	Length (m)	Core	Name	Direction	Scenarios
1745	203tfl	205tfl	2185	Yes	A3211 Victoria Embankment / Upper Thames St NEB	East	1,3
442	191tfl	184tfl	899	Yes	A201 Farringdon / A201 Blackfriars Bridge SEB	South	1,2,3
445	194tfl	185tfl	938	No	A3200 Southwark St / A300 Southwark Bridge Rd NEB	N/A	1,2,3
453	184tfl	191tfl	893	Yes	A201 Blackfriars Bridge / Farringdon Rd NWB	North	1,2,3
461	185tfl	194tfl	1076	Yes	A300 Southwark Bridge / A3200 Southwark St SWB	N/A	1,2
1620	185tfl	001tfl	1269	No	A3211 Southwark Bridge to Tower Hill SEB	N/A	2
1453	205tfl	001tfl	1165	Yes	A3211 Lower Thames St SEB	East	2
444	001tfl	185tfl	1308	No	A3211 Lower Thames St NWB	N/A	2
462	193tfl	194tfl	1011	No	A3201 Marshalsea Rd / A3200 Southwark St NWB	N/A	2

In terms of the other model parameters, each of the models exhibits best performance with w=3 and σ=1. The performance is superior in the testing phase in terms of RMSE and NRMSE, but slightly inferior in terms of the MAPE and MASEAvg, although the models outperform the historical average in each case. Figures 7.11 a)-c) and 7.12 a)-c) show the observed versus forecast values for each of the scenarios in the training and testing phases respectively. In the training phase, the model is able to effectively model the increased UTT values that occurred on Saturday 21st, Sunday 22nd and Sunday 29th May, as well as the UTTs during the rest of the period that were closer to average levels.

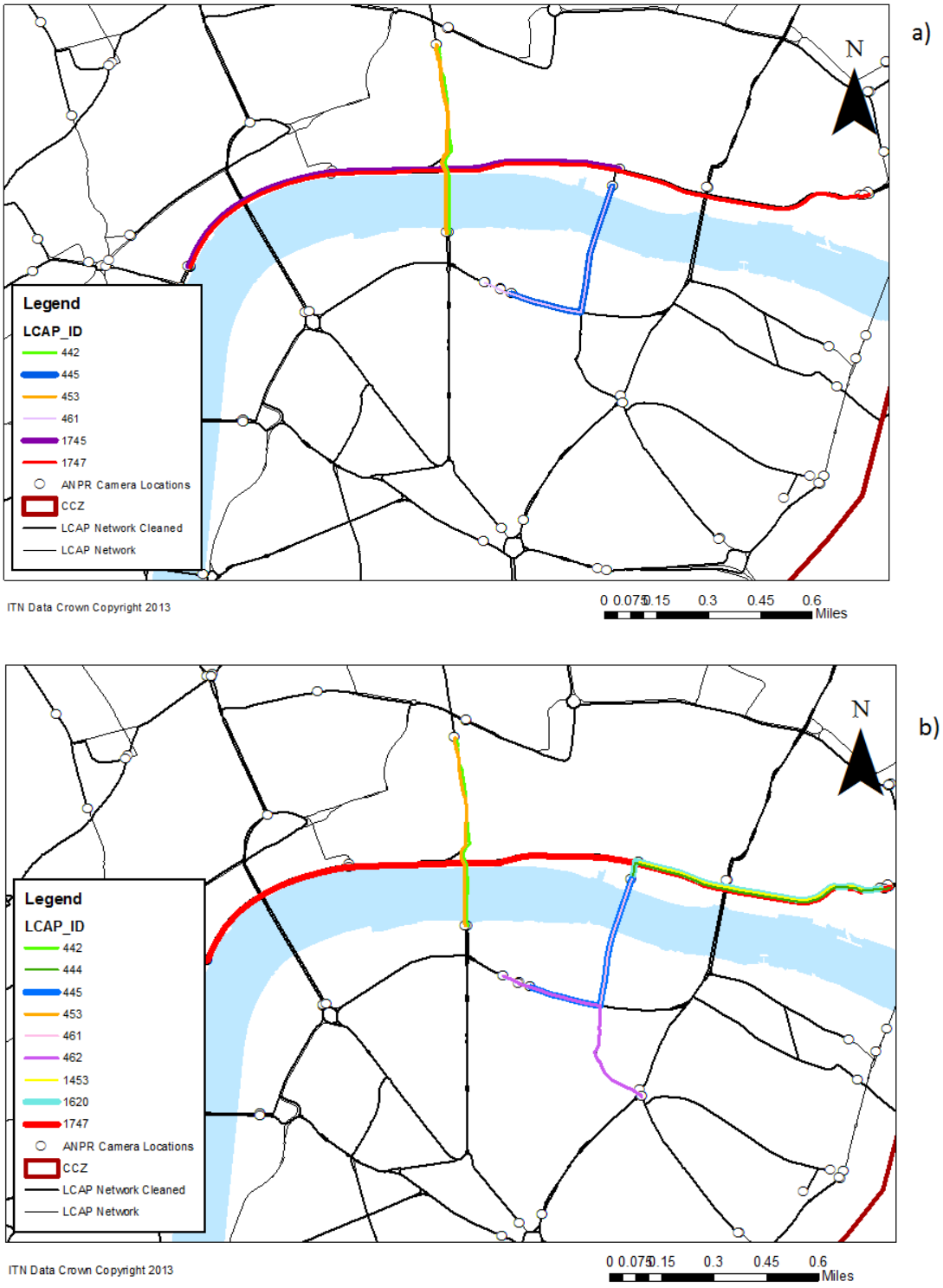


Figure 7.10 – Selected neighbours for link 1747: a) scenario 1 (k=5); b) scenario 2 (k=8)

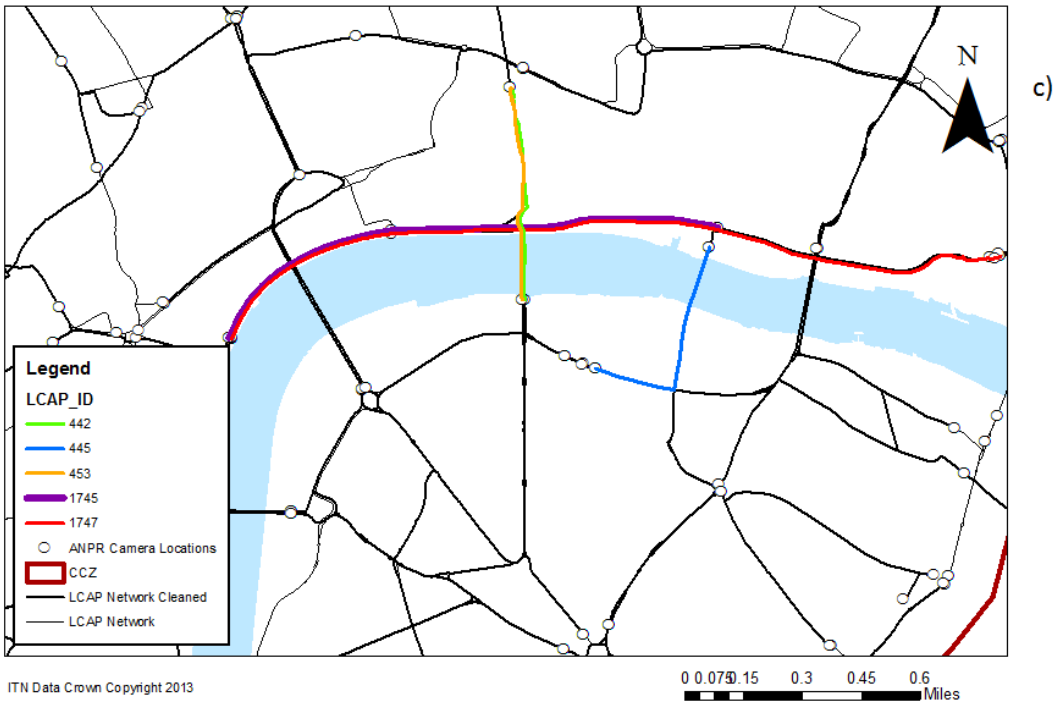
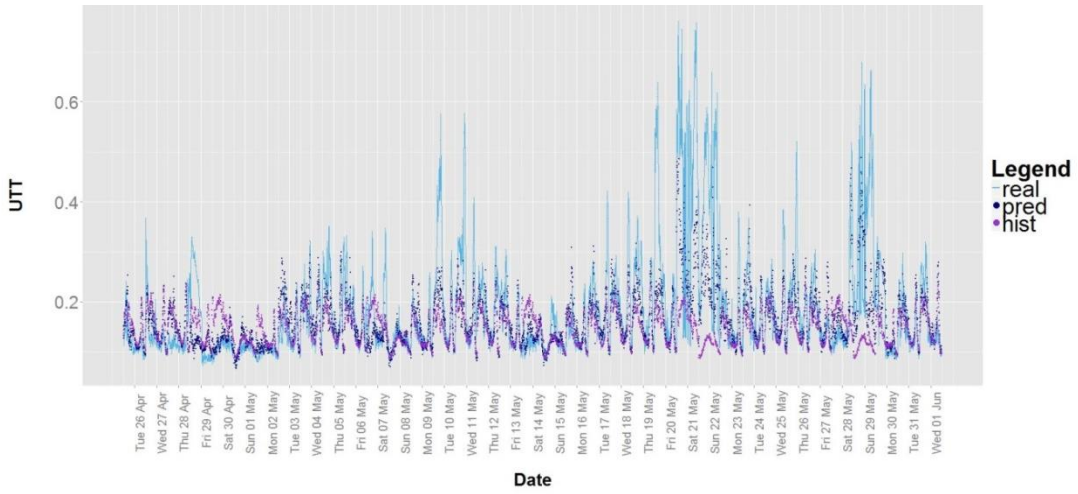


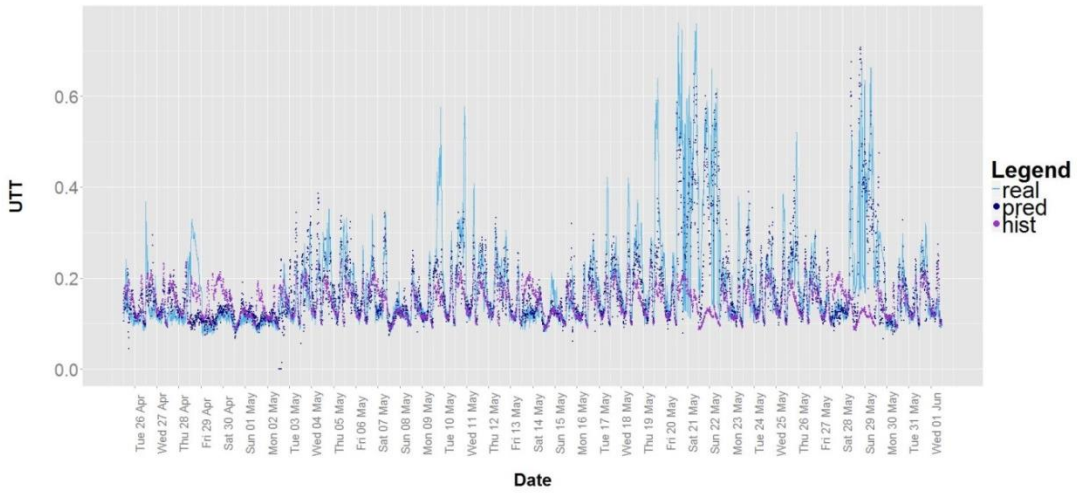
Figure 7.10 (continued) – Selected neighbours for link 1747: c) scenario 3 (k=4)

This performance is mirrored in the testing phase. On Saturday 18th June, travel times were unusually high for most of the day, representing a non-recurrent congestion event. Each of the models was able to forecast this non-recurrent event to varying degrees of success. Under scenario 2, the forecast of this congestion is very accurate considering the missing data scenario. However, some congestion events are over or under forecast by each of the models. For example, none of the models are able to accurately forecast the abnormally high travel times on Sunday 26th June. The implication of this is that the congestion was spatially isolated and therefore not forecastable from the available data. Each of the models produce forecasts that are close to the historical average on this occasion, so there is no performance loss over using the historical average alone.

a)



b)



c)

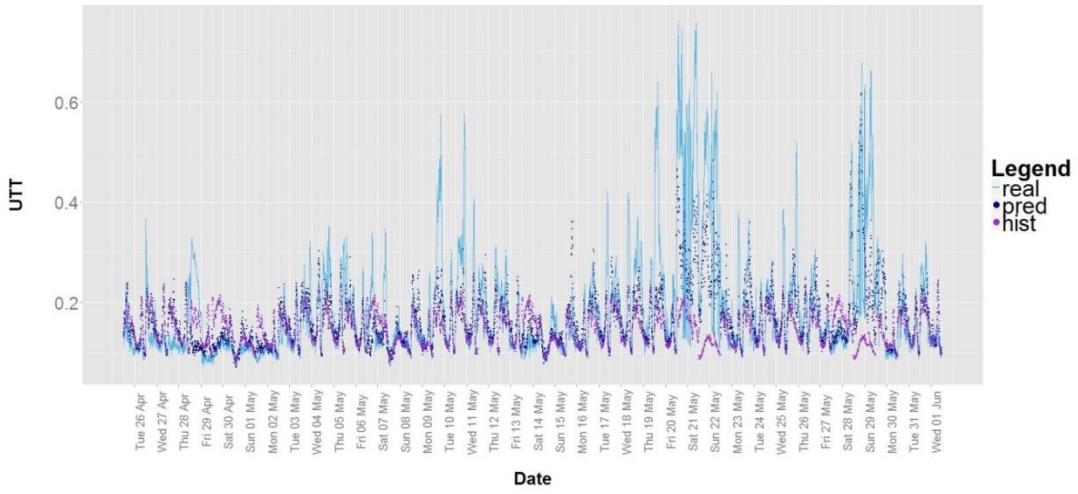


Figure 7.11 – Real versus fitted values for link Link 1747 for; a) scenario 1; b) scenario 2; and c) scenario 3.

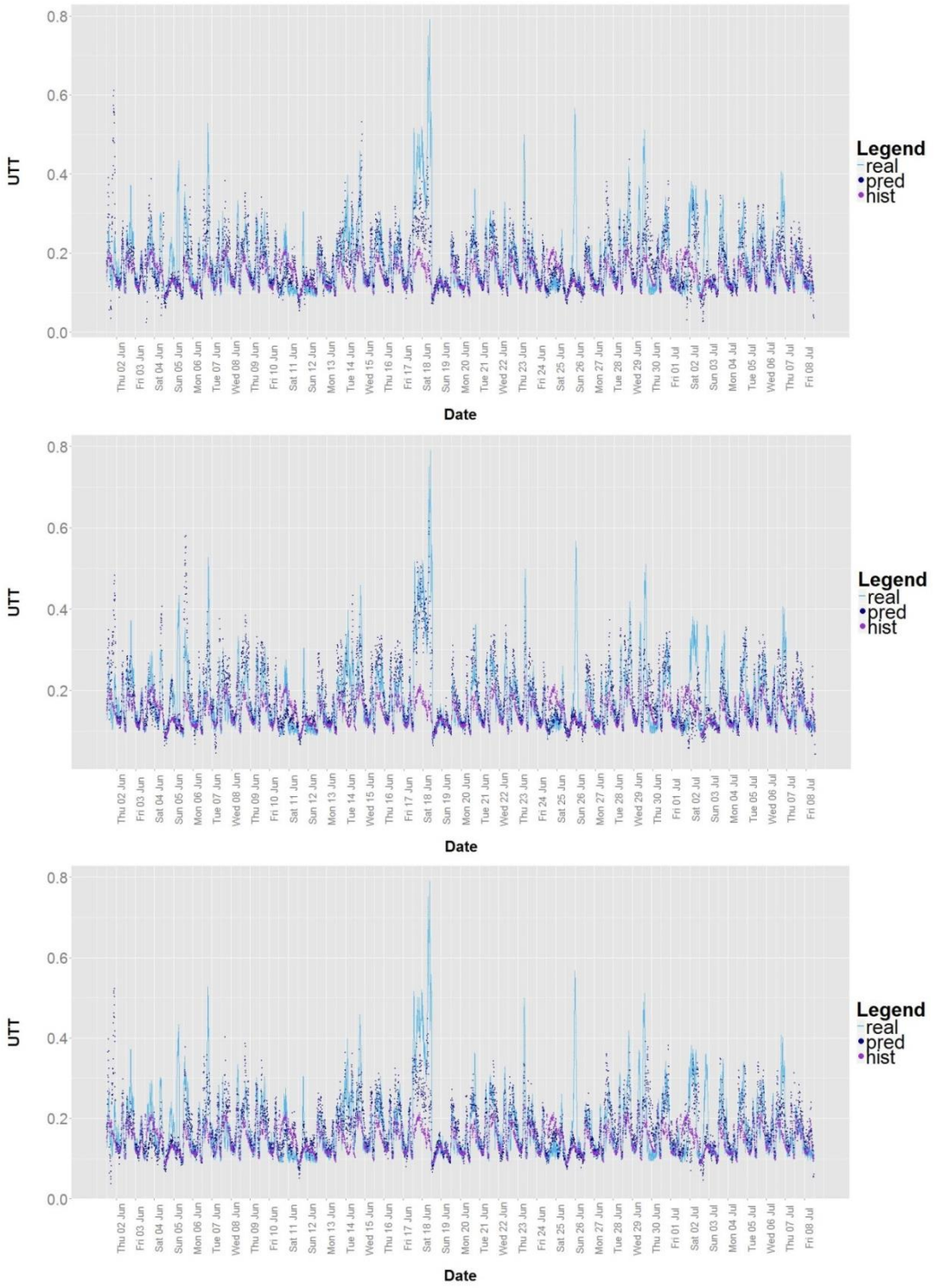


Figure 7.12 – Real versus forecast values for link 1747 for; a) scenario 1; b) scenario 2; and c) scenario 3.

7.8 Chapter Summary

In this chapter, the LOKRR model has been extended to incorporate the spatio-temporal structure of network data. An adjacency structure has been defined that combines the sensor network with the road network, which is used to define a distance measure for the network. This structure is used to train spatio-temporal models for forecasting travel times for three missing data scenarios. Two models have been tested, one in which the model parameters are fixed globally, and one in which they were allowed to vary by time of day. The forecasting results of the global model are strong, outperforming the historical average method and the *k*-NN benchmark model in the majority of cases in both the training and testing phases. The strength of the model lies in its use of the local STN to define the pattern set, which means that forecasts can only be made based on conditions that have been previously observed at the same location during the same time window. This restricts the forecasts to realistic values. The effect of this is that, where spatio-temporal dependency is found between a link and its neighbours, the model is able to forecast conditions with good accuracy. Where negligible spatio-temporal dependency is found, the model tends towards the historical average, and does not break down completely.

The performance of the local model is significantly poorer than that of the global model because the training process is not able to capture the true dependency between locations and times. This is because the LCAP network is characterised by low spatial correlation, meaning that it is difficult to identify clear and consistent spatio-temporal relationships between locations. The local model is sensitive to unusual traffic conditions or noise in the training data, which leads to overfitting. Consequently the training performance is not mirrored in the testing phase. The global model is more robust to this. However, it should be noted that the global model is only global in terms of the training process, and that the pattern set and kernel bandwidths are local due to the model structure.

Chapter 8 Model Evaluation and Extensions

8.1 Chapter Overview

In the concluding section of Chapter 5, some model requirements of a space-time model of network data were suggested. These model requirements were divided into two categories: 1) Generic model requirements for a space-time model of network data, and; 2) application specific model requirements for forecasting of road network data. In this chapter, the (ST)LOKRR model is evaluated based on these model requirements. The Chapter is organised as follows: Firstly, in section 8.2, a model evaluation framework is constructed based on the model requirements. Then, in section 8.3, the outcomes of the thesis are evaluated against in the context of this framework. In section 8.4, some potential application areas of the model are discussed. Some limitations of the methodology are described in section 8.5, and model extensions are suggested that could overcome these limitations in future research. Finally, in section 8.5, the chapter is summarised.

8.2 Model evaluation framework

In Chapter 5 of this thesis, some generic and application specific model requirements were described for the case of spatio-temporal network data, and urban road network data in particular.

Generic Requirements

1 **Accuracy and reliability:** The model should be able to forecast with an *acceptable degree of accuracy* on a *consistent basis*. The term *acceptable* is ambiguous. In this context it is evaluated based on two criteria: 1) performance compared with a naïve model, and; 2) performance compared with commonly used benchmark models.

2 **Dealing with missing and corrupt data in real time:** There are a range of methods for imputing missing data after the event, but a forecasting model should be *robust to the presence of missing data* in real time so as to maintain accuracy and reliability when sensors fail.

3 **Forecasting sufficiently far into the future so as to be fit for purpose:** The model should be capable of *accurate and reliable forecasts multiple steps into the future*, as required by the application.

4 **Maintaining computational efficiency for implementation on large networks in real time:** The model should be applicable to large networks in real time.

5 **Dealing with the temporally and spatially varying characteristics of network data, and adapting to changes in the distribution of the data:** A network model should be able to model the spatially and temporally varying characteristics of data, both in the short term and the long term, in order to ensure accuracy and reliability are maintained.

6 **Being as simple to understand and implement for practitioners as possible:** The model should be understandable, and intuitive to use if possible.

Application Specific Requirements

7 **Dealing with data that are MNAR:** The model should be able to deal with extended periods of missing data, including during peak times when accurate forecasts are most important.

8 **Forecasting for ATIS:** The model should be able to forecast travel times up to 1 hour ahead with reasonable accuracy, in order to be applicable to ATIS and other systems.

9 **Applicable to city wide forecasting:** The model should be capable of operating in real time on the whole LCAP network (~100+ links).

10 **Dealing with sparsely located sensor network and weak spatio-temporal CC:** The model should accommodate the spatial structure of the LCAP network, and model the heterogeneity in the spatio-temporal relationship between locations. Furthermore, it should be capable of producing reasonable forecasts in the presence of weak CC.

Based on these requirements, a model evaluation framework is devised here that condenses the requirements into 5 criteria against which the performance of a model of network data can be evaluated. These criteria are not specific to the transport application, but the generic and application specific requirements they cover are indicated in brackets.

a. **Exhibits forecasting performance that is at least comparable with methods already described in the literature, multiple steps into the future (1, 3 and 8).**
b. **Is capable of modelling the temporally and spatially varying characteristics of network data, if necessary (5 and 10).**
c. **Can account for the common occurrence of missing and corrupt data (2 and 7).**

d. **Is suitable for application to large networks in real time (4 and 9).**
e. **Is as simple to understand and implement for practitioners as possible (6).**

8.3 Evaluation of the proposed method against the model evaluation framework

In this section, the proposed model is evaluated against each of the criteria (a-e) defined in section 8.2 in turn.

8.3.1 (a) Comparable performance with existing methods, multiple time steps into the future

The broad remit of this research was to create a forecasting model of network data that is able to make reliable forecasts multiple time steps into the future. In order to be viable, the model must perform at least comparably with other forecasting models that are commonly used for the same tasks. In this context, model performance is taken to mean forecasting accuracy, as measured by a statistical error measure. Five error measures have been used in this thesis to assess forecasting accuracy, which are the RMSE, NRMSE, MAPE, MASE and MASEAvg. Based on evaluation of forecasting accuracy using these measures, the (ST)LOKRR model outperforms the selected benchmark models in the majority of cases described in Chapters 6 and 7. Therefore, within the remit of this thesis, the proposed model has achieved the objective of being a viable forecasting model. However, it must be acknowledged that there is a wealth of other forecasting models with which the performance of the LOKRR model could be compared, and a range of other performance measures that could be employed. The comparison models used here have been selected based on a review of the traffic forecasting literature only, and a more detailed review of the literature in other application areas may lead to different comparison models being selected.

One of the difficulties in comparing the performance of a forecasting model with the state of the art is that it is difficult to determine what model represents the state of the art when different forecasting models are applied under different conditions. This problem has been acknowledged in the time series literature for some time, and there are various time series that are publically available for benchmarking, such as the well-known Santa Fe dataset (Weigend and Gershenfeld, 1994). Keogh and Kasetty (2003) highlight the importance of benchmarking in the time series literature, noting that forecasting models should ideally be

applied to many different datasets. The model proposed here has only been applied to a single dataset and as such there is a risk of data bias. However, it has been acknowledged that, in its current form, the model is only applicable to series that display similar seasonal characteristics to travel times.

8.3.1.1 Comparison with SVR

A more in depth comparison between LOKRR and SVR is warranted here because of the similarity of the two algorithms. The improvement in performance of LOKRR over SVR has important implications for pattern based forecasting models of traffic data and possibly of seasonal time series in general. The SVR benchmark models use kernels of size 12600*12600. The computational burden of constructing kernels of this size is very high, particularly in the context of network wide forecasting. SVR has the advantage of a sparse solution because the kernel only needs to be constructed in the training phase, after which the solution depends on the support vectors. However, it is often found that a large number of data points are used as support vectors in SVR. Table 8.1 shows the number of support vectors in the trained models for each of the ten test links. With the exception of link 1815, the minimum percentage of data patterns used as support vectors in the trained models is 49.9%. Generally the percentage of support vectors is over 60%.

Table 8.1 – Number of support vectors for each of the SVR models

Link	15 minute	30 minute	45 minute	60 minute
24	9286	9639	9785	9975
26	10068	12600	10533	12599
442	12598	9232	9380	9445
453	10573	10745	10812	12441
454	9393	9758	12331	12600
881	8480	8976	9230	9286
1798	6292	11886	7952	8455
1799	9489	9817	9858	10090
1815	1544	1928	2327	2267
2448	9132	9595	9818	9886

In the LOKRR models, the maximum size of kernel that is evaluated at any time point is 560*560, which is around 2.75 times less than the smallest number of support vectors in the trained SVR models. Consequently, although the KRR algorithm is not sparse, the local nature of the model means that a much smaller amount of data can be used to obtain results that are at least comparable with SVR. By accessing only the most relevant data patterns, LOKRR is able to retain a longer memory using a smaller kernel.

KRR also has the distinct advantage over SVR that it is based on linear least squares regression, which means that it is potentially easier to understand than SVR for those without a strong background in machine learning. It is also simpler to implement. However, it should be noted that the local temporal window approach used here is not limited to LOKRR. It is straightforward to train the same model structure with a different learning algorithm. For example, online SVR could replace OKRR. Alternatively, one could apply k-NN regression or standard kernel regression in the same framework. Finally, it should be acknowledged that examples of local kernel based models exist in other application areas, and the LOKRR model described here should be compared with them to further assess its broader impact.

8.3.2 (b) Modelling the temporally and spatially varying characteristics of network data

It has been well documented in the literature that network data, and traffic data in particular, exhibit characteristics that vary in time and space. The LOKRR model defined here allows the model parameters to vary in time and space, enabling it to deal with spatial heterogeneity and temporal dynamics. However, it was found in Chapter 7 that a spatially and temporally local parameter specification resulted in overfitting. Therefore, this thesis has not fully met this criterion. It is possible that the reason for this is related to the lack of strong spatial dependence in the data, with weak CC being found in Chapter 5. On network datasets with higher spatial resolution, it is possible that more consistent correlations would be found. The presence of weak CC enhances the impact of the results of Chapter 7, where strong performance was exhibited under missing data, despite low CC.

It should be noted that, although the parameter specification in the LOKRR and STLOKRR is global, the model structure is still local as the parameters are based on the data around each time point. Thus, the estimates of the parameters still vary by time of day. This allows the model to yield improved performance over the single bandwidth SVR model despite using

smaller kernels. Moreover, the model is online, allowing it to adapt to changes in the data distribution without retraining.

8.3.3 (c) Forecasting performance under missing data

Missing data has been recognised as an important issue in many spatio-temporal datasets. In Chapter 7, the STLOKRR model has been applied to forecasting of travel times under missing data. The results demonstrate that reasonable accuracy can be achieved in most cases, even where the spatial and temporal gaps in data collection are large. Although this criterion has been broadly met, the model only outperforms the historical average on between 58.8% and 70% of cases in the three scenarios tested, so there is definite room for improvement. Furthermore, although the model implicitly deals with MNAR data by considering a 100% missing rate, it does not explicitly incorporate knowledge of the nature of the missing data into the algorithm. Its performance has also not been assessed with varying degrees of missing data.

8.3.4 (d) Large scale modelling of network data in real time

In Chapter 3, it was noted that urban traffic forecasting typically focusses on networks of a few links. In Chapter 6 of this thesis, the proposed LOKRR model was applied to the forecasting of travel times on a network of 80 links. The training of the STLOKRR model (and the comparison models) implemented in this thesis was carried out using the UCL *Legion* High Performance Computing Facility (Legion@UCL), which enables many parameter combinations to be tested simultaneously, thus greatly reducing the training time. At present, the algorithm is unsuitable for serial implementation, so this criterion has not been fully met. However, the model has the embarrassingly parallel property (Harris et al., 2010), meaning it can be adapted to exploit GPU or cloud based parallel architecture. This would enable it to scale to larger networks. The key to the scalability of the algorithm is in its local structure. A single model is trained at each location, independent of all other locations.

8.3.5 (e) Simplicity of implementation

Simplicity of implementation is subjective, and depends on the experience of the user. However, since LOKRR has its roots in OLS, it may be considered more understandable and easier to implement than other ML models such as ANNs and SVR. This is because OLS is typically taught in high school or undergraduate mathematics classes, as well as in many other subjects such as economics, geography, sociology and the sciences. Furthermore,

because the parameters are interpretable, training an LOKRR model is straightforward. For applications to large datasets, one can set a desired level of accuracy, and search within a reduced parameter space whilst guaranteeing reasonable accuracy. However, it should be noted that similarly complex parameter selection methods to those used in other models, such as genetic algorithms (Üstün et al., 2005), particle swarm optimisation (Y. Wang et al., 2009) and ant colony optimisation (Zheng et al., 2008), may be used to train LOKRR models. This decision depends on the priorities and experience of the practitioner.

8.4 Application in intelligent transportation systems

In this section, the potential application areas for the (ST)LOKRR model within ITS are discussed. The application areas can be divided into two categories, which are advanced traveller information systems (ATIS) and urban traffic management and control (UTMC) systems. These are discussed in turn in the following subsections.

8.4.1 Advance traveller information systems

In this thesis, the development of the (ST)LOKRR model has been focussed on application within ATIS. ATIS is "the systematic application of information and communications technologies to the collection of travel-related data and the processing and delivery of information of value to the traveller" (McQueen et al., 2002, p.32). The primary function of ATIS is to provide information to the traveller that will improve their journey. This may involve, for example, taking a different, less congested route between A and B, or changing departure time in order to avoid heavy traffic. There is a diverse range of platforms for ATIS, including satellite navigation systems and routing devices in vehicle; variable message signs (VMS) on the roadside; and journey planners that can be accessed before departure. In the current state of the art, ATIS makes use of live traffic data to provide tailored routes based on current conditions, and to warn users of potential disruptions to their journey. However, these systems do not currently make good use of predictive information.

The (ST)LOKRR model described in this thesis could be used within ATIS to provide more accurate estimates of potential travel times when the planned departure time is in the future, for example, up to one hour ahead as presented in Chapter 6. This would enable commuters to adjust their departure time on a daily basis and to take alternative routes when necessary, which may lead to higher journey time reliability across the network, a key goal of both transport authorities and of individual travellers (Chen et al., 2003b; Z. Li et al.,

2010). However, crucial to the effectiveness of such a system would be modelling the effect of predictive information provision on traveller behaviour (Khattak et al., 1996). Currently, the market penetration of intelligent routing systems is too low for it to have a measurable effect on network performance. However, if a sufficient number of travellers follow the same routing advice, the advised route become congested while the original route clears. This is a problem of balancing user optimum with system optimum (LeBlanc and Abdulaal, 1984). The effect of this could be investigated by integrating the STLOKRR model with traffic simulation models such as those that were briefly introduced in Chapter 3.

This problem is particularly suited to ABM. Dia (2002) propose such a model, in which a behavioural survey of drivers is used to incorporate commuters' responses to traveller information in an ABM. The ABM is implemented within a microscopic traffic simulation tool. Wahle et al. (2002) propose a similar approach, arguing that it is necessary to integrate models of knowledge-based behaviour into traffic flow models. Incorporating such models with data driven forecasting models enables updating of forecasts based on models of response to predictive information. In the context of this study, traffic simulation models are not applicable due to data sparsity, but as more data are collected and the situation improves, data driven forecasting and traffic simulation will be increasingly complementary to one another.

8.4.2 Urban traffic management and control systems

Currently, predictive information is used within UTMC for dynamic signal control (DSC) in systems such as SCOOT (Hunt et al., 1981). Typically, the forecasts made within DSC are very short term (seconds rather than minutes), and are used to model the flow between adjacent intersections as well as queue lengths and travel times (Gartner and Stamatiadis, 1998). Although (ST)LOKRR can be used for short term forecasts, its structure is tailored towards longer term forecasts, where it makes use of knowledge of the relationship between the current traffic state and the spatially and temporally local history.

Within UTMC, (ST)LOKRR is more suited to application at the operational level for corridor management. Forecasts can be used for proactive management of corridors through updating variable message signs and adjusting signal timings at the macro level. However, as noted in section 8.4.1, it is important to model the likely effect of such information on future network conditions.

8.5 Model Extensions

In this section, a number of extensions and improvements to the (ST)LOKRR model are described. In doing so, the limitations of the methodology are elucidated. These limitations are summarized around the following points:

1. The addition and removal of data points from the kernels is arbitrary.
2. The model cannot forecast states that are not described by the kernels.
3. The neighbourhood selection approach assumes correlation decreases linearly with network distance.
4. Improving missing data treatment
5. Incorporation of other data sources
6. Application to other networks

Some methods for addressing each of these limitations are given in the following subsections.

8.5.1 Online sparsification of kernels

In the present form of LOKRR, $2w+1$ data patterns are added and removed from the kernels at each time point, where w is the window size, with the oldest patterns always being removed and the newest patterns being added regardless of their position in the feature space. This is a reasonable approach in time series applications as one may consider that near patterns are more likely to be useful than distant patterns in time, particularly if the series is nonstationary. However, the approach does not consider the importance of particular types of pattern, or the redundancy of new patterns. Improvements may be realised if a test were employed to determine whether or not a newly observed pattern should be added to the kernel, and if so, which pattern should be removed from the current kernel. The kernel recursive least squares (KRLS) algorithm adopts this methodology (Engel et al., 2004). Van Vaerenbergh et al. (2010) also propose an extension to the regularized KRLS algorithm that allows data points to be added and removed from the kernel in a principled way based on a discarding criterion. This framework is extended in (Van Vaerenbergh et al., 2012). Online sparsification may allow smaller kernels to be defined, and yield improved performance in non-recurrent conditions.

8.5.2 Anomaly detection

One of the drawbacks of the LOKRR model is that its accuracy relies on each of the kernels containing a sufficient richness of patterns to describe the diversity of situations that may occur. In the traffic forecasting context, this means each kernel must contain both normal and abnormal patterns, relative to the average conditions at each time point. The empirical case studies in Chapters 5 and 6 demonstrate that the local kernels capture sufficient information to be successful in forecasting many situations. However, the model, like most forecasting models, cannot deal with situations that are not described by the training data. The online nature of LOKRR allows the model structure to adapt and react to these abnormal situations after the event, but it is during the event when accurate forecasts are most important.

This is a problem of the detection and forecasting of non-recurrent events, which is inherently difficult. One way of dealing with non-recurrent events in a forecasting context is through anomaly detection, otherwise known as outlier detection and novelty detection. Comprehensive reviews can be found in Markou and Singh (2003a, 2003b), Hodge and Austin (2004), Chandola et al. (2009), and (T. Ahmed et al., 2007) review anomaly detection on networks. Kernel methods can be extended to carry out anomaly detection in a seamless way. This has been addressed in the context of kernel algorithms in other application areas. For example, the kernel-based online anomaly detection (KOAD) algorithm of (Tarem Ahmed et al., 2007) modifies the kernel recursive least squares (KRLS) algorithm of Engel et al. (2004) to enable anomaly detection in high speed backbone traffic networks. This type of approach could be combined with the online sparsification detailed in section 8.4.1 to allow LOKRR to carry out anomaly detection, sparsification and forecasting simultaneously.

8.5.3 Improving spatio-temporal neighbourhood selection

The k-nearest neighbours STN selection approach detailed in Chapter 7 assumes a linear relationship between network distance and correlation, which may be unnecessarily restrictive. There is an implicit assumption in the k-nearest neighbours method that the nearest link to the midpoint of a link will provide the most useful information. If the most informative link is a higher order neighbour, then the model will grow until it includes this link, but less informative links may be included in the model, contributing noise. A different feature selection method for determining the STN may yield better forecasting results. The simplest way to do this is to simply test all combinations of the STN and use the model that

minimises the empirical error. However, in high dimensional data this is computationally infeasible, and more sophisticated variable selection methods may aid model selection (Guyon and Elisseeff, 2003).

In the current literature, the LASSO method of Kamarianakis et al. (2012) yields good results because all variables are considered jointly in the feature selection process, but it is imited to linear threshold regression. The GLASSO method of Gao et al. (2011) allows STN selection that is independent of the model, and spatially and temporally local GLASSO could enable local spatio-temporal neighbourhood selection for any model type.

Furthermore, as the series at adjacent measurement locations are incorporated into a single input vector, their respective distances from the location of interest are not explicitly accounted for. A spatio-temporal Kriging based approach could be used to accomplish this. Zou et al. (2012) have investigated this in the context of floating vehicle data in a purely spatial approach.

8.5.4 Improving missing data treatment

In Chapter 7, a spatio-temporal LOKRR was described for forecasting under missing data, with the assumption of long term sensor failure. The proposed method does not explicitly take into account the nature of missing data. It was shown in Chapter 4 that data on the LCAP are MNAR, and the missing rate is positively correlated with the value of the observed travel time and hence with the time of day. In the proposed methodology, this relationship has not been exploited, but there are ways in which it could be used to enhance the modelling framework. For example, the absence of travel time observations could signal increased congestion or an incident. This knowledge could be exploited in two ways; firstly, the absence of an observation, or a sequence of observations, could be used as an early warning system for congestion or incidents, aiding traffic controllers who carry out incident detection by viewing ANPR data directly. Secondly, the fact that a travel time observation is related to its likelihood of being missing could be incorporated into the forecasting algorithm.

8.5.5 Incorporating other data sources

The methods developed in this thesis have been tested in a purely autoregressive setting, where only TT observations collected on the LCAP network are considered. In previous studies, TTs have been estimated using other traffic variables such as flows and signal timings (Liu, 2008). In recent years there has been a sharp increase in the collection and use of data from floating vehicles equipped with GPS devices or mobile phones. These devices generate trajectories, from which speeds and travel times can be extracted. The availability of travel time data from such sources would enable more accurate forecasting under missing data, and would also facilitate forecasting on road links that are not covered by the LCAP network. If such data become available, the LCAP dataset could be used to validate their reliability in providing accurate measures of traffic variables.

Besides the measurement of traffic variables, other data sources such as incident and weather data may also improve forecasts. The incorporation of additional variables in the LOKRR model is straightforward because it is a kernel based model. In other application areas, kernel based methods are typically applied to very high dimensional data such as gene sequences, so the dimensionality of the problems encountered here is low in comparison, even if data from many other sources were incorporated. The success of this approach would, however, rely on sufficient volume and quality of data being available, and real correlations being extracted.

8.5.6 Extension to other datasets and data types

In this thesis, the (ST)LOKRR model was applied to forecasting of TTs collected on the LCAP network. However, (ST)LOKRR could be applied without modification to the forecasting of any time series with a regular seasonal component such as electrical load (Hahn et al., 2009), internet traffic, economic time series (Ghysels et al., 2006), temperatures and retail spending patterns amongst many others. For example, the application of seasonal SVR models has been explored in electrical load forecasting (Hong, 2011b; Hong et al., 2013) using a similar seasonal adjustment approach to Hong (2012, 2011a, 2010). The method can also be applied to any spatial or network dataset, providing an appropriate distance matrix or spatial adjacency matrix has been defined.

In its current form, LOKRR is not appropriate for application to data that do not have a regular seasonal component. However, the model could be adapted to model local

characteristics of data that are not seasonal in nature. Taking the example of traffic data, traffic can be divided into distinct states such as free flowing and congested. A kernel could be defined to describe each traffic state. Two issues would need to be tackled to accomplish this. Firstly, the historical database of patterns would need to be segmented into a number of classes that each describes a certain traffic state. Secondly, new data patterns would need to be classified into one of these classes prior to a forecast being made. This type of approach has been taken in, for example, the SVM experts approach of (Cao, 2003). This type of approach may be prone to misclassification of the input patterns, so an ensemble learning procedure may help in this regard.

Although traffic data display cyclic variation and nonlinear characteristics, they can be considered trend stationary in the context of short term forecasting. Traffic levels tend to be higher in the winter months than the summer, and there is a long term trend of increasing traffic levels, but the change is very gradual. These characteristics are captured by the online nature of the algorithm, and do not have a noticeable effect on short term forecasting accuracy. The model has not been tested on series that have nonstationary trends, such as financial series, and modifications would be needed to enable this.

8.6 Chapter Summary

In this chapter, the proposed model has been evaluated against a set of evaluation criteria that can be used to determine the success of a forecasting model of network data. These criteria have been met with varying degrees of success. The LOKRR model exhibits strong performance in forecasting travel times multiple time steps into the future, and is able to accommodate spatially and temporally varying structure. Furthermore, it is online and adapts to changes in the data distribution, and exhibits reasonable performance under missing data. However, there are still significant improvements that could be realised by implementing the model extensions described in section 8.4. These will form the basis of future research.

Chapter 9 Conclusion

9.1 Chapter Overview

In this Chapter, the thesis is concluded. First, in section 9.2, a chapter by chapter summary of the thesis is provided, in which the main outcomes of the research are summarised. In section 9.3, the potential contributions of the research to the literature are considered. Finally, in section 9.4, an outlook is given for the future of research in spatio-temporal forecasting of network data.

9.2 Thesis Summary

The Introduction in Chapter 1 and literature review in Chapter 2 demonstrate that the network science literature is now well developed in terms of our understanding of network structure. However, significant challenges remain in forecasting data collected on networks. These problems arise because of the nonlinearities and heterogeneities inherent in network data, which are exacerbated by the huge volumes of data that must be processed that are often corrupted by noise and missing data. Traditional global statistical space-time models typically perform poorly in this environment and there has been a surge in research into developing better models as a consequence.

In reviewing the state of the art in traffic forecasting in Chapter 3, the spatio-temporal approaches to forecasting on road networks have been categorised. Because of their importance to society, road networks have received considerable attention and the traffic forecasting literature is well developed. The best approaches can model local/dynamic properties of network data, are robust to the presence of missing data, can forecast multiple time steps into the future, and are scalable to large networks. However, none of the existing methods are capable of fulfilling all of these criteria.

In Chapter 4, the LCAP network is introduced. The data analysis in this section reveals that the network is characterised by high levels of missing data, a finding consistent with many studies of network data. Furthermore, the sensors are sparsely located on a dense sensor network. The spatial relationship between these sensors is uncertain, and cannot be directly determined from the sensor network structure alone. In response to this, a method is developed in Chapter 5 for connecting the sensor network to the physical network. This provides a better measure of network distance between locations on the sensor network which can be used in space-time models. A space-time autocorrelation analysis is carried

out, that reveals strong, seasonal temporal autocorrelation and weak spatial cross-correlation.

The seasonal patterns present in the LCAP data are typical of congested networks, and in Chapter 6 an LOKRR model is developed that is able to model this type of data using a local kernel based approach. In the context of forecasting time series of travel times, the LOKRR model outperforms three benchmark models: 1) ARIMA; 2) Elman RNN, and; 3) SVR. A comparison of the results of LOKRR with SVR reveals that LOKRR yields accuracy that is at least comparable with, and often better than SVR, while using smaller kernels. This finding has implications for pattern based modelling of network data, suggesting that local temporal model structures can be employed that allow a smaller historical database to be accessed at each time step without decreases in forecasting accuracy. The reason for the strong performance of LOKRR is that it can capture the temporal variability in the data through its local form. An additional strength of the model is that it has its roots in linear least squares regression, and hence is simple to understand and implement.

The LOKRR model is extended to incorporate spatial structure in Chapter 7, which enables it to cope with the presence of missing data. The STLOKRR model is applied to network wide (80 links) spatio-temporal forecasting of travel times on a spatially sparse sensor network under three scenarios of missing data. The model shows promising results in forecasting travel times where there are large spatial and temporal gaps in data collection. The performance of the model compares favourably with two benchmark models: 1) historical average, and; 2) k-Nearest Neighbours. The local structure of the model is the principal reason for the strong performance, allowing the model to perform well even when the level of spatio-temporal dependence between sensor locations is low. However, the model does not perform well when trained using local parameters, and appears to overfit the training data.

An evaluation framework for forecasting models of network data is defined in Chapter 8, against which the outcomes of the research are evaluated. The criteria of the evaluation framework are broadly met. However, a number of areas for improvement are identified. Some model extensions are suggested for implementing these improvements, which will form the basis of future research.

9.3 Contributions to the literature

In this section, some potential contributions of the research contained in this thesis to the scientific literature are summarised. The contributions are divided into two areas: 1) contributions to the literature on forecasting of network data, and; 2) contributions to the transport literature.

9.3.1 Contributions to the literature on forecasting of network data

The proposed approach addresses the significant challenge of forecasting data collected on large networks by:

1. Modelling the local spatio-temporal properties of congested networks, which enables greater forecast accuracy.
2. Allowing efficient online updating in the presence of new data, thus accommodating the dynamics of network data.
3. Dealing with the presence of missing data in real time, rather than after the event, facilitating timely intervention to prevent network failure.
4. Integrating local structure, missing data treatment and online learning in a single forecasting environment.

9.3.2 Contributions to the transport literature

The proposed approach contributes to the literature on forecasting of network data by:

1. Providing an approach for forecasting urban TTs using only link based ANPR data. The basic model structure is applicable to any link based ANPR system. By tackling the issue of defining the spatial adjacency structure of the sparsely located urban sensor network, the model enables forecasting performance to be maintained in the presence of missing data, which is important in ensuring continuity in ITS.
2. Modelling the local properties of road network data in a nonlinear machine learning framework, which to date has been attempted only with linear models.
3. Providing an approach that enables reliable forecasting of TTs multiple time steps into the future, which is important in the context of ATIS.
4. Providing an approach for large scale modelling of urban transportation networks, with many sensors.

5. Providing an approach for online modelling of urban network data, allowing long term robustness to the time varying characteristics of traffic data. This approach reduces the need for periodic retraining of forecasting models.

9.4 Conclusion and outlook

In this thesis, a novel forecasting method, (ST)LOKRR has been proposed for the forecasting of spatio-temporal network data. The model has a structure that is local in space and time, and is online, allowing it to account for spatial heterogeneity and temporal dynamics. It uses spatio-temporal information in order to ensure strong performance under missing data. The model has been applied to the forecasting of travel time data on London's road network, and exhibits strong performance compared with benchmark forecasting models. In its current form, the model is applicable to forecasting of the many types of network data that have a regular seasonal pattern, such as road traffic, electrical load and internet traffic.

As the population of the Earth continues to increase, and the emerging economies of the world continue to develop apace, the networks that we all use will come under ever increasing strain. Already, many major cities of the world suffer from crippling levels of traffic congestion, airports are stretched to capacity, and electrical grids operate well beyond their limits (Romero, 2012). Growth must be managed effectively in order to avoid the systems of the world grinding to a halt. Forecasting of the data collected on these networks is just one way in which the negative effects may be minimised. A collaborative effort is required between stakeholders involved in the management and use of networks in order to ensure and increase their resilience.

Although improvements may be realised through forecasting and intelligent management, the fact remains that the fundamental problem the world's networks face is one of demand outstripping capacity. In a broader context, policies and technologies must continue to be explored to address this basic inequality. On transportation networks, measures such as congestion pricing, fuel duties and toll routes can be used to disincentivise car use and encourage green travel. However, such policies are often at odds with the demands of the populace and can lead to other social inequalities, particularly during austere times. Satisfying the needs of all parties is a difficult balancing act.

To conclude, spatio-temporal forecasting of network data is an effective tool for the management and reduction of negative externalities of over-stretched real world networks. But it is just one tool in the armoury of those responsible for managing networks. The continuing and efficient function of real-world networks requires a holistic approach, drawing on techniques and approaches from across many domains, particularly that of network science.

Bibliography

Abdulhai, B., Porwal, H., Recker, W., 2002. Short-Term Traffic Flow Prediction Using Neuro-Genetic Algorithms. J. Intell. Transp. Syst. Technol. Plan. Oper. 7, 3.

Ahmed, M.S., Cook, A.R., 1979. ANALYSIS OF FREEWAY TRAFFIC TIME-SERIES DATA BY USING BOX-JENKINS TECHNIQUES. Transp. Res. Rec.

Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H., 2010. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econom. Rev. 29, 594–621.

Ahmed, T., Coates, M., Lakhina, A., 2007. Multivariate online anomaly detection using kernel recursive least squares, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. pp. 625–633.

Ahmed, T., Oreshkin, B., Coates, M., 2007. Machine Learning Approaches to Network Anomaly Detection, in: Proc. Second Workshop Tackling Computer Systems Problems with Machine Learning (SysML07). Presented at the Second Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML07), Cambridge, MA.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control 19, 716– 723.

Albert, R., Barabási, A.L., 2002. Statistical mechanics of complex networks. Rev. Mod. Phys. 74, 47.

Albert, R., Jeong, H., Barabasi, A.-L., 2000. Error and attack tolerance of complex networks. Nature 406, 378–382.

Alfares, H.K., Nazeeruddin, M., 2002. Electric load forecasting: Literature survey and classification of methods. Int. J. Syst. Sci. 33, 23–34.

Amisigo, B.A., Van De Giesen, N.C., 2005. Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series [WWW Document]. URL http://hal.archives-ouvertes.fr/hal-00304820/ (accessed 9.26.11).

Amjady, N., 2001. Short-term hourly load forecasting using time-series modeling with peak load estimation capability. IEEE Trans. Power Syst. 16, 498–505.

Anacleto Junior, O., Queen, C., Albers, C., 2012. Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors. J. R. Stat. Soc. Ser. C Appl. Stat. (In press).

Anacleto, O., Queen, C., Albers, C.J., 2013a. Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors. J. R. Stat. Soc. Ser. C Appl. Stat. 62, 251–270.

Anacleto, O., Queen, C., Albers, C.J., 2013b. Forecasting Multivariate Road Traffic Flows Using Bayesian Dynamic Graphical Models, Splines and Other Traffic Variables. Aust. N. Z. J. Stat. 55, 69–86.

Anselin, L., 1988. Spatial econometrics: methods and models. Springer.

Anselin, L., 1995. Local Indicators of Spatial Association - LISA. Geogr. Anal. 27, 1–25.

Arnott, R., Small, K., 1994. The Economics of Traffic Congestion. Am. Sci. 82, 446–455.

Ballabio, C., 2009. Spatial prediction of soil properties in temperate mountain regions using support vector regression. Geoderma 151, 338–350.

Baltagi, B.H., 2005. Econometric analysis of panel data. John Wiley and Sons.

Baltagi, B.H., Li, D., 2006. Prediction in the Panel Data Model with Spatial Correlation: The Case of Liquor. SSRN ELibrary.

Barabási, A.-L., Albert, R., 1999. Emergence of Scaling in Random Networks. Science 286, 509 –512.

Barceló, J., 2010. Models, Traffic Models, Simulation, and Traffic Simulation, in: Barceló, J. (Ed.), Fundamentals of Traffic Simulation, International Series in Operations Research & Management Science. Springer New York, pp. 1–62.

Barceló, J., Casas, J., 2005. Dynamic Network Simulation with AIMSUN, in: Kitamura, R., Kuwahara, M. (Eds.), Simulation Approaches in Transportation Analysis, Operations Research/Computer Science Interfaces Series. Springer US, pp. 57–98.

Barth, M., Boriboonsomsin, K., 2008. Real-World Carbon Dioxide Impacts of Traffic Congestion. Transp. Res. Rec. J. Transp. Res. Board 2058, 163–171.

Barthélemy, M., 2011. Spatial networks. Phys. Rep. 499, 1–101.

Basu, S., Mukherjee, A., Klivansky, S., 1996. Time series models for internet traffic, in: Proceedings IEEE INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Presented at the Proceedings IEEE INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation, pp. 611–620 vol.2.

Beamonte, A., Gargallo, P., Salvador, M., 2008. Bayesian inference in STAR models using neighbourhood effects. Stat. Model. 8, 285 –311.

Beamonte, A., Gargallo, P., Salvador, M., 2010. Analysis of housing price by means of STAR models with neighbourhood effects: a Bayesian approach. J. Geogr. Syst. 12, 227–240.

Bell, M.G.H., Iida, Y., 1997. Transportation Network Analysis. Wiley-Blackwell.

Benoit Mandelbrot, 1963. The Variation of Certain Speculative Prices. J. Bus., Journal of Business 36.

Bergmeir, C., Benítez, J.M., 2012. Neural networks in R using the stuttgart neural network simulator: RSNNS. J. Stat. Softw. 46, 1–26.

Bhattacharya, B., Solomatine, D.P., 2005. Machine learning in soil classification, in: Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on. Presented at the Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on, pp. 2694–2699 vol. 5.

Billings, D., Yang, J.-S., 2006. Application of the ARIMA Models to Urban Roadway Travel Time Prediction - A Case Study, in: 2006 IEEE International Conference on Systems, Man, and Cybernetics. Presented at the 2006 IEEE International Conference on Systems, Man, and Cybernetics, Taipei.

Black, W.R., 1992. Network Autocorrelation in Transport Network and Flow Systems. Geogr. Anal. 24, 207–222.

Black, W.R., Thomas, I., 1998. Accidents on Belgium's motorways: a network autocorrelation analysis. J. Transp. Geogr. 6, 23–31.

Blandin, S., El Ghaoui, L., Bayen, A., 2009. Kernel regression for travel time estimation via convex optimization, in: Proceedings of the 48th IEEE Conference on Decision and Control, 2009 Held Jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Presented at the Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009, pp. 4360–4365.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D., 2006. Complex networks: Structure and dynamics. Phys. Rep. 424, 175–308.

Bogaert, P., 1996. Comparison of kriging techniques in a space-time context. Math. Geol. 28, 73–86.

Boisvert, J.B., Deutsch, C.V., 2010. Geostatistics with locally varying anisotropy. University of Alberta.

Börjesson, M., Fosgerau, M., Algers, S., 2012. Catching the tail: Empirical identification of the distribution of the value of travel time. Transp. Res. Part Policy Pract. 46, 378–391.

Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A Training Algorithm for Optimal Margin Classifiers, in: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. ACM Press, pp. 144–152.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., 1994. Time series analysis: forecasting and control. Prentice Hall.

Boyd, D., Crawford, K., 2011. Six Provocations for Big Data (SSRN Scholarly Paper No. ID 1926431). Social Science Research Network, Rochester, NY.

Brackstone, M., McDonald, M., 1999. Car-following: a historical review. Transp. Res. Part F Traffic Psychol. Behav. 2, 181–196.

Brandes, U., Robins, G., McCRANIE, A., Wasserman, S., 2013. What is network science? Netw. Sci. 1, 1–15.

Breiman, L., 2001. Statistical modeling: The two cultures. Stat. Sci. 199–215.

Browne, 2000. Cross-Validation Methods. J. Math. Psychol. 44, 108–132.

Burges, C.J.C., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. Data Min. Knowl. Discov. 2, 121–167.

Cai, Z., Li, S., Zhang, X., 2009. Tourism demand forecasting by support vector regression and genetic algorithm, in: Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on. Presented at the Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, pp. 144–146.

Cao, L., 2003. Support vector machines experts for time series forecasting. Neurocomputing 51, 321–339.

Caputo, B., Sim, K., Furesjo, F., Smola, A., 2002. Appearance-based object recognition using SVMs: which kernel should I use?, in: Proc of NIPS Workshop on Statitsical Methods for Computational Experiments in Visual Processing and Computer Vision. Presented at the NIPS workshop on Statitsical methods for computational experiments in visual processing and computer vision, Whistler.

Cardillo, A., Scellato, S., Latora, V., Porta, S., 2006. Structural properties of planar graphs of urban street patterns. Phys. Rev. E 73, 066107.

Castells, M., 2010. Globalisation, Networking, Urbanisation: Reflections on the Spatial Dynamics of the Information Age. Urban Stud. 47, 2737–2745.

Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., Han, L.D., 2009. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. 36, 6164–6173.

Cetin, N., Burri, A., Nagel, K., 2003. A large-scale agent-based traffic microsimulation based on queue model, in: IN PROCEEDINGS OF SWISS TRANSPORT RESEARCH CONFERENCE (STRC), MONTE VERITA, CH. pp. 3–4272.

Chan, K.Y., Dillon, T., Chang, E., Singh, J., 2013a. Prediction of Short-Term Traffic Variables Using Intelligent Swarm-Based Neural Networks. IEEE Trans. Control Syst. Technol. 21, 263–274.

Chan, K.Y., Dillon, T.S., Chang, E., 2013b. An Intelligent Particle Swarm Optimization for Short-Term Traffic Flow Forecasting Using on-Road Sensor Systems. IEEE Trans. Ind. Electron. 60, 4714–4725.

Chan, K.Y., Dillon, T.S., Singh, J., Chang, E., 2012a. Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg #x2013;Marquardt Algorithm. IEEE Trans. Intell. Transp. Syst. 13, 644–654.

Chan, K.Y., Khadem, S., Dillon, T.S., Palade, V., Singh, J., Chang, E., 2012b. Selection of Significant On-Road Sensor Data for Short-Term Traffic Flow Forecasting Using the Taguchi Method. IEEE Trans. Ind. Inform. 8, 255–266.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Comput. Surv. CSUR 41, 15.

Chandra, S.R., Al-Deek, H., 2008. Cross-Correlation Analysis and Multivariate Prediction of Spatial Time Series of Freeway Traffic Speeds. Transp. Res. Rec. J. Transp. Res. Board 2061, 64–76.

Chandra, S.R., Al-Deek, H., 2009. Predictions of Freeway Traffic Speeds and Volumes Using Vector Autoregressive Models. J. Intell. Transp. Syst. Technol. Plan. Oper. 13, 53.

Chatfield, C., 2004. The analysis of time series: an introduction. Chapman & Hall/CRC.

Chen, C., Kwon, J., Rice, J., Skabardonis, A., Varaiya, P., 2003a. Detecting Errors and Imputing Missing Data for Single-Loop Surveillance Systems. Transp. Res. Rec. J. Transp. Res. Board 1855, 160–167.

Chen, C., Skabardonis, A., Varaiya, P., 2003b. Travel-Time Reliability as a Measure of Service. Transp. Res. Rec. J. Transp. Res. Board 1855, 74–79.

Chen, C., Wang, Y., Li, L., Hu, J., Zhang, Z., 2012. The retrieval of intra-day trend and its influence on traffic prediction. Transp. Res. Part C Emerg. Technol. 22, 103–118.

Cheng, T., Haworth, J., Wang, J., 2011. Spatio-temporal autocorrelation of road network data. J. Geogr. Syst. 14, 389–413.

Chong, C.-Y., Kumar, S.P., 2003. Sensor networks: evolution, opportunities, and challenges. Proc. IEEE 91, 1247–1256.

Chun, Y., 2008. Modeling network autocorrelation within migration flows by eigenvector spatial filtering. J. Geogr. Syst. 10, 317–344.

Cliff, A.D., Ord, J.K., 1969. The Problem of Spatial Autocorrelation, in: Papers in Regional Science. Pion, London, pp. 25–55.

Cliff, A.D., Ord, J.K., 1975. Space-Time Modelling with an Application to Regional Forecasting. Trans. Inst. Br. Geogr. 119–128.

Cohen, R., Havlin, S., ben-Avraham, D., 2003. Efficient Immunization Strategies for Computer Networks and Populations. Phys. Rev. Lett. 91, 247901.

Cook, P.A., 1987. Nonlinear Dynamical Systems, New edition. ed. Prentice-Hall.

Cressie, N., 1988. Spatial prediction and ordinary kriging. Math. Geol. 20, 405–421.

Cressie, N., Wikle, C.K., 2002. Space-Time Kalman Filter, in: Encyclopedia of Environmetrics. John Wiley & Sons, Ltd.

Cressie, N., Wikle, C.K., 2011. Statistics for Spatio-Temporal Data. John Wiley & Sons.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, 1st ed. Cambridge University Press.

Csardi, G., Nepusz, T., 2006. The igraph software package for complex network research. InterJournal Complex Syst. 1695.

Daganzo, C.F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. Part B Methodol. 28, 269–287.

Daganzo, C.F., 1995. The cell transmission model, part II: Network traffic. Transp. Res. Part B Methodol. 29, 79–93.

De Martino, D., Dall'Asta, L., Bianconi, G., Marsili, M., 2009. Congestion phenomena on complex networks. Phys. Rev. E 79, 015101.

De Smith, M.J., Goodchild, M.F., Longley, P., 2007. Geospatial analysis: a comprehensive guide to principles, techniques and software tools. Matador, Leicester.

Dia, H., 2002. An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. Transp. Res. Part C Emerg. Technol. 10, 331–349.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.

Ding, Q.Y., Wang, X.F., Zhang, X.Y., Sun, Z.Q., 2010. Forecasting Traffic Volume with Space-Time ARIMA Model. Adv. Mater. Res. 156-157, 979–983.

Dobson, I., Carreras, B.A., Lynch, V.E., Newman, D.E., 2007. Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization. Chaos Interdiscip. J. Nonlinear Sci. 17, 026103.

Dong, W., Pentland, A., 2009. A Network Analysis of Road Traffic with Vehicle Tracking Data., in: AAAI Spring Symposium: Human Behavior Modeling. pp. 7–12.

Doreian, P., Teuter, K., Wang, C.-H., 1984. Network Autocorrelation Models: Some Monte Carlo Results. Sociol. Methods Res. 13, 155–200.

Dougherty, M., 1995. A review of neural networks applied to transport. Transp. Res. Part C Emerg. Technol. 3, 247–260.

Dougherty, M.S., Cobbett, M.R., 1997. Short-term inter-urban traffic forecasts using neural networks. Int. J. Forecast. 13, 21–31.

Dow, M., Burton, M., White, D., 1982. Network autocorrelation: A simulation study of a foundational problem in regression and survey research. Soc. Netw. 4, 169–200.

Dow, M.M., 2007. Galton's Problem as Multiple Network Autocorrelation Effects: Cultural Trait Transmission and Ecological Constraint. Cross-Cult. Res. 41, 336–363.

Dow, M.M., Burton, M.L., White, D.R., Reitz, K.P., 1984. Galton's Problem as network autocorrelation. Am. Ethnol. 11, 754–770.

Dow, M.M., Eff, E.A., 2008. Global, Regional, and Local Network Autocorrelation in the Standard Cross-Cultural Sample. Cross-Cult. Res. 42, 148 –171.

Drucker, H., Burges, C.J.C., Kaufman, L., C, C.J., Kaufman, B.L., Smola, A., Vapnik, V., 1996. Support Vector Regression Machines.

Egger, P., Pfaffermayr, M., Winner, H., 2005. An unbalanced spatial panel data approach to US state tax competition. Econ. Lett. 88, 329–335.

Elhorst, J.P., 2003. Specification and Estimation of Spatial Panel Data Models. Int. Reg. Sci. Rev. 26, 244 –268.

Elman, J.L., 1990. Finding structure in time* 1. Cogn. Sci. 14, 179–211.

Engel, Y., Mannor, S., Meir, R., 2004. The kernel recursive least-squares algorithm. IEEE Trans. Signal Process. 52, 2275–2285.

Engle, R., 2001. GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics. J. Econ. Perspect. 15, 157–168.

Engle, R.F., 1982. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica 50, 987–1007.

Ettema, D., Timmermans, H., 2006. Costs of travel time uncertainty and benefits of travel time information: Conceptual model and numerical examples. Transp. Res. Part C Emerg. Technol. 14, 335–350.

Ettema, D., Timmermans, H., 2007. Space–Time Accessibility Under Conditions of Uncertain Travel Times: Theory and Numerical Simulations. Geogr. Anal. 39, 217–240.

Exterkate, P., 2013. Model selection in kernel ridge regression. Comput. Stat. Data Anal. 68, 1–16.

Farber, S., Páez, A., Volz, E., 2009. Topology and Dependency Tests in Spatial and Network Autoregressive Models. Geogr. Anal. 41, 158–180.

Fei, X., Lu, C.-C., Liu, K., 2011. A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. Transp. Res. Part C Emerg. Technol. 19, 1306–1318.

Fellendorf, M., Vortisch, P., 2010. Microscopic Traffic Flow Simulator VISSIM, in: Barceló, J. (Ed.), Fundamentals of Traffic Simulation, International Series in Operations Research & Management Science. Springer New York, pp. 63–93.

Flahaut, B., Mouchart, M., Martin, E.S., Thomas, I., 2003. The local spatial autocorrelation and the kernel method for identifying black zones: A comparative approach. Accid. Anal. Prev. 35, 991–1004.

Florax, R.J.G.M., Rey, S., 1995. The Impacts of Misspecified Spatial Interaction in Linear Regression Models, in: Anselin, P.D.L., Florax, D.R.J.G.M. (Eds.), New Directions in Spatial Econometrics, Advances in Spatial Science. Springer Berlin Heidelberg, pp. 111–135.

Fosgerau, M., Fukuda, D., 2012. Valuing travel time variability: Characteristics of the travel time distribution on an urban road. Transp. Res. Part C Emerg. Technol. 24, 83–101.

Fotheringham, A.S., Brunsdon, C., Charlton, M., 2002. Geographically weighted regression: the analysis of spatially varying relationships. John Wiley and Sons.

Friedman, J., Hastie, T., Tibshirani, R., 2008. Sparse inverse covariance estimation with the graphical lasso. Biostat. Oxf. Engl. 9, 432–441.

Gao, Y., Sun, S., Shi, D., 2011. Network-scale traffic modeling and forecasting with graphical lasso. Adv. Neural Networks–ISNN 2011 151–158.

Garrison, W.L., 1960. Connectivity of the Interstate Highway System. Pap. Reg. Sci. 6, 121–137.

Garrison, W.L., Marble, D.F., 1962. THE STRUCTURE OF TRANSPORTATION NETWORKS,.

Gartner, N., Stamatiadis, C., 1998. Integration of Dynamic Traffic Assignment with Real-Time Traffic Adaptive Control System. Transp. Res. Rec. J. Transp. Res. Board 1644, 150–156.

Geary, R.C., 1954. The Contiguity Ratio and Statistical Mapping. Inc. Stat. 5, 115–146.

Genton, M.G., 2002. Classes of kernels for machine learning: a statistics perspective. J. Mach. Learn. Res. 2, 299–312.

Geroliminis, N., Daganzo, C.F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transp. Res. Part B Methodol. 42, 759–770.

Gething, P.W., Atkinson, P.M., Noor, A.M., Gikandi, P.W., Hay, S.I., Nixon, M.S., 2007. A local space–time kriging approach applied to a national outpatient malaria data set. Comput. Geosci. 33, 1337–1350.

Getis, A., 2009. Spatial Weight Matrices. Geogr. Anal. 41, 404–410.

Getis, A., Aldstadt, J., 2004. Constructing the Spatial Weights Matrix Using a Local Statistic. Geogr. Anal. 36, 90–104.

Getis, A., Ord, J.K., 1992. The Analysis of Spatial Association by Use of Distance Statistics. Geogr. Anal. 24, 189–206.

Ghosh, B., Basu, B., O'Mahony, M., 2009. Multivariate Short-Term Traffic Flow Forecasting Using Time-Series Analysis. IEEE Trans. Intell. Transp. Syst. 10, 246–254.

Ghysels, E., Osborn, D.R., Rodrigues, P.M.M., 2006. Chapter 13 Forecasting Seasonal Time Series, in: G. Elliott, C.W.J.G. and A.T. (Ed.), Handbook of Economic Forecasting. Elsevier, pp. 659–711.

Gilardi, N., Bengio, S., 2001. Local Machine Learning Models for Spatial Data Analysis. J. Geogr. Inf. Decis. Anal. 4, 2000.

Girosi, F., Jones, M., Poggio, T., 1995. Regularization Theory and Neural Networks Architectures. Neural Comput. 7, 219–269.

Glasbey, C.A., 1995. Imputation of missing values in spatio-temporal solar radiation data. Environmetrics 6, 363–371.

Gneiting, T., 2002. Nonseparable, Stationary Covariance Functions for Space–Time Data. J. Am. Stat. Assoc. 97, 590–600.

Goodwin, P., 2004. The economic costs of road traffic congestion (Report). UCL (University College London), The Rail Freight Group, London, UK.

Gordon, R.L., Tighe, W., 2005. Traffic Control Systems Handbook. Federal Highway Administration, Washington DC, USA.

Granger, C.W.J., 1978. An introduction to bilinear time series models. Vandenhoeck und Ruprecht.

Griffith, D.A., 2009. Spatial Autocorrelation.

Griffith, D.A., 2010. Modeling spatio-temporal relationships: retrospect and prospect. J. Geogr. Syst. 12, 111–123.

Griffith, D.A., Lagona, F., 1998. On the quality of likelihood-based estimators in spatial autoregressive models when the data dependence structure is misspecified. J. Stat. Plan. Inference 69, 153–174.

Guin, A., 2006. Travel Time Prediction using a Seasonal Autoregressive Integrated Moving Average Time Series Model, in: Proceedings of the IEEE ITSC 2006. Presented at the 2006 IEEE Intelligent Transportation Systems Conference, Toronto.

Guo, J., 2005. Adaptive estimation and prediction of univariate vehicular traffic condition series (PhD Thesis). North Carolina State University, North Carolina, US.

Guo, J., Smith, B.L., 2007. Short Term Speed Variance Forecasting Using Linear Stochastic Modeling of Univariate Traffic Speed Series (Final Report No. UVACTS-15-17-10). University of Virginia Center for Transportation Studies, Charlottesville, US.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Hahn, H., Meyer-Nieberg, S., Pickl, S., 2009. Electric load forecasting methods: Tools for decision making. Eur. J. Oper. Res. 199, 902–907.

Hamed, M.M., Al-Masaeid, H.R., Said, Z.M.B., 1995. Short-Term Prediction of Traffic Volume in Urban Arterials. J. Transp. Eng. 121, 249–254.

Hamilton, J.D., 1994. Time Series Analysis. Princeton University Press.

Han, L., Shuai, M., Xie, K., Song, G., Ma, X., 2010. Locally kernel regression adapting with data distribution in prediction of traffic flow, in: 2010 18th International Conference on Geoinformatics. Presented at the 2010 18th International Conference on Geoinformatics, pp. 1–6.

Hardoon, D.R., Szedmak, S., Shawe-Taylor, J., 2004. Canonical Correlation Analysis: An Overview with Application to Learning Methods. Neural Comput. 16, 2639–2664.

Harris, R., Singleton, A., Grose, D., Brunsdon, C., Longley, P., 2010. Grid-enabling Geographically Weighted Regression: A Case Study of Participation in Higher Education in England. Trans. GIS 14, 43–61.

Harvill, J.L., 2010. Spatio-temporal processes. Wiley Interdiscip. Rev. Comput. Stat. 2, 375–382.

Haworth, J., Cheng, T., 2012. Non-parametric regression for space–time forecasting under missing data. Comput. Environ. Urban Syst. 36, 538–550.

Haworth, J., Cheng, T., Manley, E.J., 2013. Improving forecasting under missing data on sparse spatial networks, in: Proceedings of the 12th International Conference on GeoComputation. Presented at the Geocomputation 2013, Wuhan, China.

Helbing, D., 1996. Gas-kinetic derivation of Navier-Stokes-like traffic equations. Phys. Rev. E 53, 2366.

Helbing, D., Hennecke, A., Shvetsov, V., Treiber, M., 2002. Micro- and macro-simulation of freeway traffic. Math. Comput. Model. 35, 517–547.

Heuvelink, G.B.M., Griffith, D.A., 2010. Space-Time Geostatistics for Geography: A Case Study of Radiation Monitoring Across Parts of Germany. Geogr. Anal. 42, 161–179.

Hillier, B., Iida, S., 2005. Network and Psychological Effects in Urban Movement, in: Cohn, A.G., Mark, D.M. (Eds.), Spatial Information Theory, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 475–490.

Hodge, V.J., Austin, J., 2004. A Survey of Outlier Detection Methodologies. Artif. Intell. Rev. 22, 85–126.

Hoerl, A.E., Kennard, R.W., 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics 12, 55–67.

Hofleitner, A., Herring, R., Bayen, A., 2012. Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning. Transp. Res. Part B Methodol. 46, 1097–1122.

Holme, P., 2003. CONGESTION AND CENTRALITY IN TRAFFIC FLOW ON COMPLEX NETWORKS. Adv. Complex Syst. 06, 163–176.

Hong, W.-C., 2010. Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting. Neural Comput. Appl.

Hong, W.-C., 2011a. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. Neurocomputing 74, 2096–2107.

Hong, W.-C., 2011b. Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. Energy 36, 5568–5578.

Hong, W.-C., 2012. Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting. Neural Comput. Appl. 21, 583–593.

Hong, W.-C., Dong, Y., Zhang, W.Y., Chen, L.-Y., K. Panigrahi, B., 2013. Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. Int. J. Electr. Power Energy Syst. 44, 604–614.

Hong, W.-C., Dong, Y., Zheng, F., Lai, C.-Y., 2011. Forecasting urban traffic flow by SVR with continuous ACO. Appl. Math. Model. 35, 1282–1291.

Hoogendoorn, S.P., Bovy, P.H.L., 2001. State-of-the-art of vehicular traffic flow modelling. Proc. Inst. Mech. Eng. Part J. Syst. Control Eng. 215, 283–303.

Hotelling, H., 1936. Relations Between Two Sets of Variates. Biometrika 28, 321–377.

Howell, D.C., 2007. The treatment of missing data, in: Outhwaite, W., Turner, S.P. (Eds.), The SAGE Handbook of Social Science Methodology. SAGE Publications Ltd, London, UK.

Hsieh, W.W., 2009. Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels, 1st ed. Cambridge University Press.

Hu, C., Xie, K., Song, G., Wu, T., 2008. Hybrid Process Neural Network based on Spatio-Temporal Similarities for Short-Term Traffic Flow Prediction, in: Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on. Presented at the Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on, pp. 253–258.

HU, X., WANG, W., SHENG, H., 2010. Urban Traffic Flow Prediction with Variable Cell Transmission Model. J. Transp. Syst. Eng. Inf. Technol. 10, 73–78.

Huang, B., Wu, B., Barry, M., 2010. Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices. Int. J. Geogr. Inf. Sci. 24, 383–401.

Huang, H.-C., Cressie, N., 1996. Spatio-temporal prediction of snow water equivalent using the Kalman filter. Comput. Stat. Data Anal. 22, 159–175.

Huang, R., Sun, S., Liu, Y., 2011. Sparse kernel regression for traffic flow forecasting, in: Proceedings of the 8th International Conference on Advances in Neural Networks - Volume Part II, ISNN'11. Springer-Verlag, Berlin, Heidelberg, pp. 76–84.

Hunt, P.B., Robertson, D.I., Bretherton, R.D., Winton, R.I., 1981. SCOOT - A TRAFFIC RESPONSIVE METHOD OF COORDINATING SIGNALS. Publ. Transp. Road Res. Lab.

Hyndman, R.J., Khandakar, Y., 2007. Automatic time series for forecasting: the forecast package for R.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. Int. J. Forecast. 22, 679–688.

Imtech, 2013. SCOOT - The world's leading adaptive traffic control system [WWW Document]. URL http://www.scoot-utc.com/ (accessed 10.22.13).

Ishak, S., Alecsandru, C., 2004. Optimizing Traffic Prediction Performance of Neural Networks under Various Topological, Input, and Traffic Condition Settings. J. Transp. Eng. 130, 452–465.

Jiang, B., 2007. A topological pattern of urban street networks: universality and peculiarity. Phys. Stat. Mech. Its Appl. 384, 647–655.

Jiang, B., 2009. Street hierarchies: a minority of streets account for a majority of traffic flow. Int. J. Geogr. Inf. Sci. 23, 1033–1048.

Jin, X., Cheu, R.L., Srinivasan, D., 2002. Development and adaptation of constructive probabilistic neural network in freeway incident detection. Transp. Res. Part C Emerg. Technol. 10, 121–147.

Jin, X., Zhang, Y., Yao, D., 2007. Simultaneously prediction of network traffic flow based on PCA-SVR, in: Advances in Neural Networks–ISNN 2007. Springer, pp. 1022–1031.

Jones, S.L., Sullivan, A.J., Cheekoti, N., Anderson, M.D., Malave, D., 2004. Traffic simulation software comparison study. University Transportation Center for Alabama.

Jordan, M., 1986. Attractor dynamics and parallellism in a connectionist sequential machine. Presented at the Proceedings of the Eighth Annual Meeting of the Cognitive Science Society, Lawrence Erlbaum Associates, pp. 531–546.

Juszczak, P., Tax, D., Duin, R.P.W., 2002. Feature scaling in support vector data description, in: Proc. ASCI. pp. 95–102.

Kaastra, I., Boyd, M., 1996. Designing a neural network for forecasting financial and economic time series. Neurocomputing 10, 215–236.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng. 82, 35–45.

Kamarianakis, Y., Kanas, A., Prastacos, P., 2005. Modeling Traffic Volatility Dynamics in an Urban Network. Transp. Res. Rec. 1923, 18–27.

Kamarianakis, Y., Oliver Gao, H., Prastacos, P., 2010. Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions. Transp. Res. Part C Emerg. Technol. 18, 821–840.

Kamarianakis, Y., Prastacos, P., 2005. Space-time modeling of traffic flow. Comput. Geosci. 31, 119–133.

Kamarianakis, Y., Shen, W., Wynter, L., 2012. Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO. Appl. Stoch. Models Bus. Ind. 28, 297–315.

Kanevski, M., 2008. Advanced Mapping of Environmental Data. Wiley-ISTE.

Kanevski, M., Canu, S., 2000. Spatial Data Mapping with Support Vector Regression. IDIAP Res. Rep. RR-00-09.

Kanevski, M., Maignan, M., 2004. Analysis and modelling of spatial environmental data. EPFL Press.

Kanevski, M., Parkin, R., Pozdnukhov, A., Timonin, V., Maignan, M., Demyanov, V., Canu, S., 2004. Environmental data mining and modeling based on machine learning algorithms and geostatistics. Environ. Model. Softw. 19, 845–855.

Kanevski, M., Pozdnoukhov, A., Timonin, V., Maignan, M., 2007. Mapping of environmental data using kernel-based methods. Rev. Int. Géomat. 17, 309–331.

Kanevski, M., Timonin, V., Pozdnukhov, A., 2009. Machine Learning for Spatial Environmental Data: Theory, Applications, and Software, Har/Cdr. ed. EFPL Press.

Karatzoglou, A., Meyer, D., Hornik, K., 2006. Support vector machines in R. J. Stat. Softw. 15, 1–28.

Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A., 2004. kernlab-An S4 package for kernel methods in R.

Kariniotakis, G.N., Stavrakakis, G.S., Nogaret, E.F., 1996. Wind power forecasting using advanced neural networks models. IEEE Trans. Energy Convers. 11, 762–767.

Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transp. Res. Part C Emerg. Technol. 19, 387–399.

Kendall, M.G., Ord, J.K., 1990. Time Series. E. Arnold.

Keogh, E., Kasetty, S., 2003. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Min. Knowl. Discov. 7, 349–371.

Khattak, A., Polydoropoulou, A., Ben-Akiva, M., 1996. Modeling Revealed and Stated Pretrip Travel Response to Advanced Traveler Information Systems. Transp. Res. Rec. J. Transp. Res. Board 1537, 46–54.

Khondker, H.H., 2011. Role of the New Media in the Arab Spring. Globalizations 8, 675–679.

Kirby, H.R., Watson, S.M., Dougherty, M.S., 1997. Should we use neural networks or statistical models for short-term motorway traffic forecasting? Int. J. Forecast. 13, 43–50.

Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection 1137–1143.

Kokaram, A., Godsill, S., 1996. A system for reconstruction of missing data in image sequences using sampled 3D AR models and MRF motion priors. Comput. Vision—ECCV96 613–624.

Kokaram, A., Godsill, S., 1997. Joint detection, interpolation, motion and parameter estimation for image sequences with missing data, in: Image Analysis and Processing. pp. 719–726.

Kokaram, A.C., 2004. On missing data treatment for degraded video and film archives: a survey and a new Bayesian approach. IEEE Trans. Image Process. 13, 397–415.

Kokaram, A.C., Godsill, S.J., 2002. MCMC for joint noise reduction and missing data treatment in degraded video. IEEE Trans. Signal Process. 50, 189–205.

Kokaram, A.C., Morris, R.D., Fitzgerald, W.J., Rayner, P.J.., 1995a. Detection of missing data in image sequences. IEEE Trans. Image Process. 4, 1496–1508.

Kokaram, A.C., Morris, R.D., Fitzgerald, W.J., Rayner, P.J.., 1995b. Interpolation of missing data in image sequences. IEEE Trans. Image Process. 4, 1509–1519.

Kotsialos, A., Papageorgiou, M., 2001. The Importance of Traffic Flow Modeling for Motorway Traffic Control. Netw. Spat. Econ. 1, 179–203.

Kotsialos, A., Papageorgiou, M., Diakaki, C., Pavlis, Y., Middelham, F., 2002. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. IEEE Trans. Intell. Transp. Syst. 3, 282–292.

Kotusevski, G., Hawick, K.A., 2009. A review of traffic simulation software.

Kriegel, H.-P., Renz, M., Schubert, M., Züfle, A., 2012. Efficient Traffic Density Prediction in Road Networks Using Suffix Trees. KI - Künstl. Intell. 26, 233–240.

Krige, d g, 1951. A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand.

Kuhn, M., 2008. Building predictive models in R using the caret package. J. Stat. Softw. 28, 1–26.

Kwon, J., Mauch, M., Varaiya, P., 2006. Components of Congestion: Delay from Incidents, Special Events, Lane Closures, Weather, Potential Ramp Metering Gain, and Excess Demand. Transp. Res. Rec. J. Transp. Res. Board 1959, 84–91.

Kyriakidis, P.C., Journel, A.G., 1999. Geostatistical space–time models: a review. Math. Geol. 31, 651–684.

Lahiri, S.K., Ghanta, K.C., 2008. The Support Vector Regression with the parameter tuning assisted by a differential evolution technique: Study of the critical velocity of a slurry flow in a pipeline. Association of Chemical Engineers.

Larose, D.T., 2004. Discovering Knowledge in Data: An Introduction to Data Mining, 1st ed. Wiley-Interscience.

LeBlanc, L.J., Abdulaal, M., 1984. A comparison of user-optimum versus system-optimum traffic assignment in transportation network design. Transp. Res. Part B Methodol. 18, 115–121.

Lee, S., Fambro, D., 1999. Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting. Transp. Res. Rec. 1678, 179–188.

Lee, T.-H., White, H., Granger, C.W.J., 1993. Testing for neglected nonlinearity in time series models : A comparison of neural network methods and alternative tests. J. Econom. 56, 269–290.

Leenders, R.T.A.., 2002. Modeling social influence through network autocorrelation: constructing the weight matrix. Soc. Netw. 24, 21–47.

Leng, Z., Gao, J., Qin, Y., Liu, X., Yin, J., 2013. Short-term forecasting model of traffic flow based on GRNN, in: Control and Decision Conference (CCDC), 2013 25th Chinese. Presented at the Control and Decision Conference (CCDC), 2013 25th Chinese, pp. 3816–3820.

Levin, M., Tsao, Y.-D., 1980. ON FORECASTING FREEWAY OCCUPANCIES AND VOLUMES (ABRIDGMENT). Transp. Res. Rec.

Li, J., Cai, Z., 2008. A Novel Automatic Parameters Optimization Approach Based on Differential Evolution for Support Vector Regression. Adv. Comput. Intell. 510–519.

Li, L., Li, Y., Li, Z., 2013. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. Transp. Res. Part C Emerg. Technol. 34, 108–120.

Li, M.-W., Hong, W.-C., Kang, H.-G., 2013. Urban traffic flow forecasting using Gauss–SVR with cat mapping, cloud model and PSO hybrid algorithm. Neurocomputing 99, 230–240.

Li, W., Yang, Y., 2008. Hybrid kernel learning via genetic optimization for TS fuzzy system identification. Int. J. Adapt. Control Signal Process. n/a–n/a.

Li, X., Shao, M., Ding, L., Xu, G., Li, J., 2010. Particle Swarm Optimization-based LS-SVM for Building Cooling Load Prediction. J. Comput. 5.

Li, Z., Hensher, D.A., Rose, J.M., 2010. Willingness to pay for travel time reliability in passenger transport: A review and some new empirical evidence. Transp. Res. Part E Logist. Transp. Rev. 46, 384–403.

Lighthill, M.J., Whitham, G.B., 1955. On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. Proc. R. Soc. Lond. Ser. Math. Phys. Sci. 229, 317–345.

Liu, H., 2008. Travel Time Prediction for Urban Networks (PhD Thesis). Delft University of Technology, Delft, Netherlands.

Liu, H., van Lint, H., van Zuylen, H., Zhang, K., 2006a. Two distinct ways of using kalman filters to predict urban arterial travel time, in: Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE. Presented at the Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE, pp. 845–850.

Liu, H., van Zuylen, H., van Lint, H., Salomons, M., 2006b. Predicting Urban Arterial Travel Time with State-Space Neural Networks and Kalman Filters. Transp. Res. Rec. J. Transp. Res. Board 1968, 99–108.

Liu, Z., Sharma, S., Datla, S., 2008. Imputation of Missing Traffic Data during Holiday Periods. Transp. Plan. Technol. 31, 525–544.

Long, J., Gao, Z., Zhao, X., Lian, A., Orenstein, P., 2011. Urban Traffic Jam Simulation Based on the Cell Transmission Model. Netw. Spat. Econ. 11, 43–64.

Lotan, G., Graeff, E., Ananny, M., Gaffney, D., Pearce, I., Boyd, D., 2011. The revolutions were tweeted: Information flows during the 2011 Tunisian and Egyptian revolutions. Int. J. Commun. 5, 1375–1405.

Lynch, C., 2008. Big data: How do your data grow? Nature 455, 28–29.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A., 2011. Big data: The next frontier for innovation, competition, and productivity.

Markou, M., Singh, S., 2003a. Novelty detection: a review—part 1: statistical approaches. Signal Process. 83, 2481–2497.

Markou, M., Singh, S., 2003b. Novelty detection: a review—part 2:: neural network based approaches. Signal Process. 83, 2499–2521.

McQueen, B., Schuman, R., Chen, K., 2002. Advanced Traveler Information Systems. Artech House.

Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transp. Res. Part C Emerg. Technol. 19, 606–616.

Min, X., Hu, J., Chen, Q., Zhang, T., Zhang, Y., 2009a. Short-term traffic flow forecasting of urban network based on dynamic STARIMA model. IEEE, pp. 1–6.

Min, X., Hu, J., Chen, Q., Zhang, T., Zhang, Y., 2009b. Short-term traffic flow forecasting of urban network based on dynamic STARIMA model, in: Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on. Presented at the Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on, pp. 1–6.

Min, X., Hu, J., Zhang, Z., 2010a. Urban traffic network modeling and short-term traffic flow forecasting based on GSTARIMA model, in: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. Presented at the Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pp. 1535–1540.

Min, X., Hu, J., Zhang, Z., 2010b. Urban traffic network modeling and short-term traffic flow forecasting based on GSTARIMA model. IEEE, pp. 1535–1540.

Mitchell, T.M., 1997. Machine Learning, 1st ed. McGraw-Hill Science/Engineering/Math.

Mizruchi, M.S., Neuman, E.J., 2008. The effect of density on the level of bias in the network autocorrelation model. Soc. Netw. 30, 190–200.

Moran, P.A.P., 1950. Notes on Continuous Stochastic Phenomena. Biometrika 37, 17–23.

Mountrakis, G., Im, J., Ogole, C., 2010. Support vector machines in remote sensing: A review. ISPRS J. Photogramm. Remote Sens.

Müller, K.-R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V., 1997. Predicting time series with support vector machines, in: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D. (Eds.), Artificial Neural Networks — ICANN'97, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 999–1004.

Nadaraya, E.A., 1964. On Estimating Regression. Theory Probab. Its Appl. 9, 141.

Nagel, K., Schreckenberg, M., 1992. A cellular automaton model for freeway traffic. J. Phys. I 2, 2221–2229.

Neuman, E.J., Mizruchi, M.S., 2010. Structure and bias in the network autocorrelation model. Soc. Netw. 32, 290–300.

Newman, M.E.J., 2003. The structure and function of complex networks. SIAM Rev. 45, 167–256.

Ni, D., Leonard, J.D., Guin, A., Feng, C., 2005. A Multiple Imputation Scheme for Overcoming the Missing Values and Variability Issues in ITS Data. ASCE J. Transp. Eng. 131, 931–938.

Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transp. Res. Part B Methodol., Transportation Research Part B: Methodological 18, 1–11.

Olden, J.D., Neff, B.D., 2001. Cross-correlation bias in lag analysis of aquatic time series. Mar. Biol. 138, 1063–1070.

Ozaki, T., 1977. On the Order Determination of ARIMA Models. J. R. Stat. Soc. Ser. C Appl. Stat. 26, 290–301.

Páez, A., Scott, D.M., Volz, E., 2008. Weight matrices for social influence analysis: An investigation of measurement errors and their effect on model identification and estimation quality. Soc. Netw. 30, 309–317.

Papagiannaki, K., Taft, N., Zhang, Z.-L., Diot, C., 2003. Long-term forecasting of Internet backbone traffic: observations and initial models, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. Presented at the INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, pp. 1178–1188 vol.2.

Park, D.C., El-Sharkawi, M.A., Marks, R.J., Atlas, L.E., Damborg, M.J., 1991. Electric load forecasting using an artificial neural network. IEEE Trans. Power Syst. 6, 442–449.

Pastor-Satorras, R., Vespignani, A., 2002. Immunization of complex networks. Phys. Rev. E 65, 036104.

Pastor-Satorras, R., Vespignani, A., 2005. Epidemics and immunization in scale-free networks, in: Bornholdt, S., Schuster, H.G. (Eds.), Handbook of Graphs and Networks. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 111–130.

Payne, H.J., 1971. Models of Freeway Traffic and Control. Simulation Councils, Incorporated.

Peeters, D., Thomas, I., 2009. Network Autocorrelation. Geogr. Anal. 41, 436–443.

Pfeifer, P.E., Deutsch, S.J., 1980. A Three-Stage Iterative Procedure for Space-Time Modelling. TECHNOMETRICS 22, 35–47.

Pfeiffer, P.E., Deutsch, S.J., 1980. JSTOR: Transactions of the Institute of British Geographers, New Series, Vol. 5, No. 3 (1980), pp. 330-349. Trans. Inst. Br. Geogr., New Series 5, 330–349.

Pflieger, G., Rozenblat, C., 2010. Introduction. Urban Networks and Network Theory: The City as the Connector of Multiple Networks. Urban Stud. 47, 2723–2735.

Poggio, T., Girosi, F., 1990. Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks. Science 247, 978–982.

Pozdnoukhov, A., Kanevski, M., 2006. Multi-scale support vector regression for automatic mapping and spatial novelty detection, in: Geophysical Research Abstracts. p. 01221.

PTV Group, 2014. PTV Visum [WWW Document]. URL http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/ (accessed 3.2.14).

Qu, L., Li, L., Zhang, Y., Hu, J., 2009. PPCA-Based Missing Data Imputation for Traffic Flow Volume: A Systematical Approach. IEEE Trans. Intell. Transp. Syst. 10, 512–522.

Queen, C.M., Albers, C.J., 2008. Forecasting traffic flows in road networks: A graphical dynamic model approach., in: Proceedings of the 28th International Symposium of Forecasting, International Institute of Forecasters.

Queen, C.M., Wright, B.J., Albers, C.J., 2007. Eliciting a directed acyclic graph for a multivariate time series of vehicle counts in a traffic network. Aust. N. Z. J. Stat. 49, 221–239.

Rasmussen, C.E., Williams, C., 2006. Gaussian processes for machine learning. MIT Press, Cambridge, MA.

Richards, P.I., 1956. Shock Waves on the Highway. Oper. Res. 4, 42–51.

Rifkin, R., Yeo, G., Poggio, T., 2003. Regularized least-squares classification. Nato Sci. Ser. Sub Ser. III Comput. Syst. Sci. 190, 131–154.

Robinson, S., Polak, J., 2006. Overtaking rule method for the cleaning of matched license-plate data. J. Transp. Eng. 132, 609–617.

Romero, J.J., 2012. Blackouts illuminate India's power problems. IEEE Spectr. 49, 11–12.

Rosas-Casals, M., Valverde, S., Solé, R.V., 2007. TOPOLOGICAL VULNERABILITY OF THE EUROPEAN POWER GRID UNDER ERRORS AND ATTACKS. Int. J. Bifurc. Chaos 17, 2465–2475.

Saltyte, L., 2005. Modified STARIMA model for space-time data. Liet Matem Rink 45, 375–379.

Saunders, C., Gammerman, A., Vovk, V., 1998. Ridge regression learning algorithm in dual variables, in: (ICML-1998) Proceedings of the 15th International Conference on Machine Learning. pp. 515–521.

Schneider, T., 2001. Analysis of Incomplete Climate Data: Estimation of Mean Values and Covariance Matrices and Imputation of Missing Values. J. Clim. 14, 853–871.

Schölkopf, B., Burges, C.J.C., Smola, A.J., 1998. Advances in Kernel Methods: Support Vector Learning. The MIT Press.

Schölkopf, B., Burges, C.J.C., Smola, A.J., 1999. Advances in Kernel Methods: Support Vector Learning. MIT Press.

Schölkopf, B., Smola, A., Müller, K.-R., 1997. Kernel principal component analysis, in: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D. (Eds.), Artificial Neural Networks — ICANN'97, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 583–588.

Schölkopf, B., Smola, A.J., 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Sharma, S., Lingras *, P., Zhong, M., 2004. Effect of missing values estimations on traffic parameters. Transp. Plan. Technol. 27, 119–144.

Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA.

Shen, G., 1997. A fractal dimension analysis of urban transportation networks. Geogr. Environ. Model. 1, 221–236.

Shi, W., Fisher, P., Goodchild, M.F., 2003. Spatial Data Quality. CRC Press.

Shoesmith, G.L., 2013. Space–time autoregressive models and forecasting national, regional and state crime rates. Int. J. Forecast. 29, 191–201.

Sitharam, T.G., Samui, P., Anbazhagan, P., 2008. Spatial Variability of Rock Depth in Bangalore Using Geostatistical, Neural Network and Support Vector Machine Models. Geotech. Geol. Eng. 26, 503–517.

Skabardonis, A., Varaiya, P., Petty, K., 2003. Measuring Recurrent and Nonrecurrent Traffic Congestion. Transp. Res. Rec. J. Transp. Res. Board 1856, 118–124.

Smith, B., Demetsky, M., 1996. Multiple-Interval Freeway Traffic Flow Forecasting. Transp. Res. Rec. J. Transp. Res. Board 1554, 136–141.

Smith, B.L., Demetsky, M.J., 1997. Traffic Flow Forecasting: Comparison of Modeling Approaches. J. Transp. Eng. 123, 261–266.

Smith, B.L., Williams, B.M., Keith Oswald, R., 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. Transp. Res. Part C Emerg. Technol. 10, 303–321.

Smith, R.L., Kolenikov, S., Cox, L.H., 2003. Spatio-Temporal Modeling of PM2.5 Data with Missing Values. J. Geophys. Res. – ATMOSPHERES 128, 10–1029.

Smith, T.M., Reynolds, R.W., Livezey, R.E., Stokes, D.C., 1996. Reconstruction of Historical Sea Surface Temperatures Using Empirical Orthogonal Functions. J. Clim. 9, 1403–1420.

Smola, A.J., Sch\ölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14, 199–222.

Sonnenburg, S., R\ätsch, G., Sch\äfer, C., Sch\ölkopf, B., 2006. Large scale multiple kernel learning. J. Mach. Learn. Res. 7, 1565.

Specht, D.F., 1991. A general regression neural network. Neural Netw. IEEE Trans. On 2, 568–576.

Srinivasan, D., Wai Chan, C., Balaji, P.G., 2009. Computational intelligence-based congestion prediction for a dynamic urban street network. Neurocomputing 72, 2710–2716.

Stathopoulos, A., Karlaftis, M., 2001. Temporal and Spatial Variations of Real-Time Traffic Data in Urban Areas. Transp. Res. Rec. J. Transp. Res. Board 1768, 135–140.

Stathopoulos, A., Karlaftis, M.G., 2003. A multivariate state space approach for urban traffic flow modeling and prediction. Transp. Res. Part C Emerg. Technol. 11, 121–135.

Stetzer, F., 1982. Specifying weights in spatial forecasting models: the results of some experiments. Environ. Plan. A 14, 571 – 584.

Sun, S., Chen, Q., 2008. Kernel Regression with a Mahalanobis Metric for Short-Term Traffic Flow Forecasting, in: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2008. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 9–16.

Sun, S., Zhang, C., Yu, G., Lu, N., Xiao, F., 2004. Bayesian network methods for traffic flow forecasting with incomplete data. Mach. Learn. ECML 2004 419–428.

Sun, S., Zhang, C., Zhang, Y., 2005. Traffic flow forecasting using a spatio-temporal bayesian network predictor. Artif. Neural Netw. Form. Models Their Appl.-ICANN 2005 273–278.

Sun, Z., Arentze, T., Timmermans, H., 2005. Modeling the Impact of Travel Information on Activity-Travel Rescheduling Decisions Under Conditions of Travel Time Uncertainty. Transp. Res. Rec. J. Transp. Res. Board 1926, 79–87.

Swanson, N.R., White, H., 2011. A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks. Rev. Econ. Stat. 79, 540–550.

Szkuta, B.R., Sanabria, L.A., Dillon, T.S., 1999. Electricity price short-term forecasting using artificial neural networks. IEEE Trans. Power Syst. 14, 851–857.

Tibshirani, R., 1996. Regression Shrinkage and Selection via the Lasso. J. R. Stat. Soc. Ser. B Methodol. 58, 267–288.

Tikhonov, A.N., 1943. On the Stability of Inverse Problems. Comptes Rendus Dokl. Acad. Sci. URSS 39.

Tobler, W.R., 1970. A Computer Movie Simulating Urban Growth in the Detroit Region. Econ. Geogr. 46, 234–240.

Tonkin, E., Pfeiffer, H.D., Tourte, G., 2012. Twitter, information sharing and the London riots? Bull. Am. Soc. Inf. Sci. Technol. 38, 49–57.

Tsay, R.S., 1986. Nonlinearity tests for time series. Biometrika 73, 461.

TSS, 2014. Aimsun [WWW Document]. URL http://www.aimsun.com/wp/ (accessed 3.2.14).

U.S. Dept. of Energy, 2009. National Electric Transmission Congestion Study. U.S. Dept. of Energy.

Üstün, B., Melssen, W.J., Oudenhuijzen, M., Buydens, L.M.C., 2005. Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. Anal. Chim. Acta 544, 292–305.

Van Lint, H., Hoogendoorn, S., van Zuylen, H., 2002. State space neural networks for freeway travel time prediction. Artif. Neural Networks—ICANN 2002 135–135.

Van Lint, J.W.C., 2006. Reliable Real-Time Framework for Short-Term Freeway Travel Time Prediction. J. Transp. Eng. 132, 921–932.

Van Lint, J.W.C., 2008. Online Learning Solutions for Freeway Travel Time Prediction. Intell. Transp. Syst. IEEE Trans. On 9, 38–47.

Van Lint, J.W.C., Hoogendoorn, S.P., van Zuylen, H.J., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. Transp. Res. Part C Emerg. Technol. 13, 347–369.

Van Vaerenbergh, S., Lazaro-Gredilla, M., Santamaria, I., 2012. Kernel Recursive Least-Squares Tracker for Time-Varying Regression. IEEE Trans. Neural Netw. Learn. Syst. 23, 1313–1326.

Van Vaerenbergh, S., Santamaria, I., Liu, W., Principe, J.C., 2010. Fixed-budget kernel recursive least-squares, in: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP). Presented at the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1882–1885.

Van Vaerenbergh, S., Via, J., Santamana, I., 2006. A sliding-window kernel RLS algorithm and its application to nonlinear channel identification, in: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on. pp. V–V.

Vanajakshi, L., Rilett, L.R., 2004. A Comparison Of The Performance Of Artificial. Neural Networks And Support Vector Machines For The Prediction Of Traffic Speed, in: 2004 IEEE Intelligent Vehicles Symposium. Parma.

Vanajakshi, L., Rilett, L.R., 2007. Support Vector Machine Technique for the Short Term Prediction of Travel Time, in: Proceedings of the 2007 IEEE Intelligent Vehicles Symposium. Istanbul.

Vapnik, V., 1999. The Nature of Statistical Learning Theory, 2nd ed. Springer.

Vapnik, V., Lerner, A., 1963. Pattern recognition using generalized portrait method. Autom. Remote Control 24, 774–780.

Vapnik, V.N., Chervonenkis, A.Y., 1971. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. Theory Probab. Its Appl. 16, 264.

Vlahogianni, E.I., 2007. Prediction of non-recurrent short-term traffic patterns using genetically optimized probabilistic neural networks. Oper. Res. 7, 171–184.

Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G., 2004. Short-term traffic forecasting: Overview of objectives and methods. Transp. Rev. Transnatl. Transdiscipl. J. 24, 533.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transp. Res. Part C Emerg. Technol. 13, 211–234.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2006. Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume. Transp. Res. Part C Emerg. Technol. 14, 351–367.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2007. Spatio-Temporal Short-Term Urban Traffic Volume Forecasting Using Genetically Optimized Modular Networks. Comput.-Aided Civ. Infrastruct. Eng. 22, 317–325.

Wahle, J., Bazzan, A.L.C., Klügl, F., Schreckenberg, M., 2002. The impact of real-time information in a two-route scenario using agent-based simulation. Transp. Res. Part C Emerg. Technol. 10, 399–417.

Walters, A.A., 1961. The Theory and Measurement of Private and Social Cost of Highway Congestion. Econometrica 29, 676–699.

Walton, J.T., 2008. Subpixel urban land cover estimation: comparing cubist, random forests, and support vector regression. Photogramm. Eng. Remote Sens. 74, 1213–1222.

Wang, J., Cheng, T., Heydecker, B.G., Haworth, J., 2010. STARIMA for journey time prediction in London [WWW Document]. Proc. 5th IMA Conf. Math. Transp. URL http://discovery.ucl.ac.uk/739110/ (accessed 9.30.13).

Wang, J., Wang, Y., Zhang, C., Du, W., Zhou, C., Liang, Y., 2009. Parameter Selection of Support Vector Regression Based on a Novel Chaotic Immune Algorithm, in: Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on. Presented at the Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on, pp. 652–655.

Wang, J., Zou, N., Chang, G.-L., 2008. Travel Time Prediction: Empirical Analysis of Missing Data Issues for Advanced Traveler Information System Applications. Transp. Res. Rec. J. Transp. Res. Board 2049, 81–91.

Wang, Y., Wang, J., Du, W., Zhang, C., Zhang, Y., Zhou, C., 2009. Parameters optimization of support vector regression based on immune particle swarm optimization algorithm, in: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC '09. ACM, New York, NY, USA, pp. 997–1000.

Watson, G.S., 1964. Smooth regression analysis. Sankhyā Indian J. Stat. Ser. A 26, 359–372.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of /`small-world/' networks. Nature 393, 440–442.

Weigend, A.S., Gershenfeld, N.A., 1994. Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, Reading, MA.

White, H., 1980. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. Econometrica 48, 817–838.

Whitlock, M.E., Queen, C.M., 2000. Modelling a traffic network with missing data. J. Forecast. 19, 561–574.

Whittaker, J., Garside, S., Lindveld, K., 1997. Tracking and predicting a network traffic process. Int. J. Forecast. 13, 51–61.

Wikle, C.K., Berliner, L.M., Cressie, N., 1998. Hierarchical Bayesian space-time models. Environ. Ecol. Stat. 5, 117–154.

Wikle, C.K., Cressie, N., 1999. A Dimension-Reduced Approach to Space-Time Kalman Filtering. Biometrika 86, 815–829.

Williams, B., Durvasula, P., Brown, D., 1998. Urban Freeway Traffic Flow Prediction: Application of Seasonal Autoregressive Integrated Moving Average and Exponential Smoothing Models. Transp. Res. Rec. 1644, 132–141.

Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. J. Transp. Eng. ASCE 129, 664–672.

Williams, C.K.I., Rasmussen, C.E., 1996. Gaussian processes for regression, in: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (Eds.), Advances in Neural Information Processing Systems 8. MIT.

Willmott, C.J., Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim. Res. 30, 79.

Wooldridge, J.M., 2002. Econometric Analysis of Cross Section and Panel Data. MIT Press.

Wu, C.-H., Ho, J.-M., Lee, D.T., 2004. Travel-Time Prediction With Support Vector Regression. IEEE Trans. Intell. Transp. Syst. 5, 276–281.

Wu, C.-H., Tzeng, G.-H., Lin, R.-H., 2009. A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. Expert Syst. Appl. Int. J. 36, 4725–4735.

Xie, F., Levinson, D., 2007. Measuring the Structure of Road Networks. Geogr. Anal. 39, 336–356.

Xie, Y., Zhang, Y., Ye, Z., 2007. Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition. Comput.-Aided Civ. Infrastruct. Eng. 22, 326–334.

Xie, Y., Zhao, K., Sun, Y., Chen, D., 2010. Gaussian Processes for Short-Term Traffic Volume Forecasting. Transp. Res. Rec. J. Transp. Res. Board 2165, 69–78.

Xu, Z., Sui, D.Z., 2007. Small-world characteristics on transportation networks: a perspective from network autocorrelation. J. Geogr. Syst. 9, 189–205.

Yao, J., Tan, C.L., 2000. A case study on using neural networks to perform technical forecasting of forex. Neurocomputing 34, 79–98.

Yu, Y.J., Cho, M.-G., 2008. A Short-Term Prediction Model for Forecasting Traffic Information Using Bayesian Network, in: 2008 Third International Conference on Convergence and Hybrid Information Technology. Presented at the 2008 Third International Conference on Convergence and Hybrid Information Technology (ICCIT), Busan, Korea, pp. 242–247.

Yue, Y., Yeh, A.G.., 2008. Spatiotemporal traffic-flow dependency and short-term traffic forecasting. Environ. Plan. B Plan. Des. 35, 762–771.

Zacarias, O.P., Andersson, M., 2011. Spatial and temporal patterns of malaria incidence in Mozambique. Malar. J. 10, 189.

Zealand, C.M., Burn, D.H., Simonovic, S.P., 1999. Short term streamflow forecasting using artificial neural networks. J. Hydrol. 214, 32–48.

Zhang, Y., Liu, Y., 2009a. Missing Traffic Flow Data Prediction using Least Squares Support Vector Machines in Urban Arterial Streets, in: IEEE Symposium on Computational Intelligence and Data Mining, 2009.

Zhang, Y., Liu, Y., 2009b. Data Imputation Using Least Squares Support Vector Machines in Urban Arterial Streets. Signal Process. Lett. IEEE 16, 414–417.

Zhang, Y., Xie, Y., 2008. Forecasting of Short-Term Freeway Volume with *v*-Support Vector Machines. Transp. Res. Rec. J. Transp. Res. Board 2024, 92–99.

Zhang, Y., Yang, X., 2008. Error and Attack Tolerance Topological Analysis of Urban Road Networks, in: The Eighth International Conference of Chinese Logistics and Transportation Professionals. Presented at the Proceedings of the Eighth International Conference of Chinese Logistics and Transportation Professionals, Chengdu, China, pp. 575–575.

Zheng, L., Yu, M., Yu, S., 2008. Support Vector Regression and Ant Colony Optimization for Combustion Performance of Boilers, in: Natural Computation, 2008. ICNC '08. Fourth International Conference on. Presented at the Natural Computation, 2008. ICNC '08. Fourth International Conference on, pp. 178–182.

Zhong, M., Lingras, P., Sharma, S., 2004a. Estimation of missing traffic counts using factor, genetic, neural, and regression techniques. Transp. Res. Part C Emerg. Technol. 12, 139–166.

Zhong, M., Sharma, S., Lingras, P., 2004b. Genetically Designed Models for Accurate Imputation of Missing Traffic Counts. Transp. Res. Rec. 1879, 71–79.

Zhong, M., Sharma, S., Lingras, P., 2006. Matching Patterns for Updating Missing Values of Traffic Counts. Transp. Plan. Technol. 29, 141–156.

Zhu, T., Kong, X., Lv, W., 2009. Large-scale Travel Time Prediction for Urban Arterial Roads Based on Kalman Filter, in: Internationa Conference on Computational Intelligence and Software Engineering, 2009.

Zou, H., Yue, Y., Li, Q., Yeh, A.G.O., 2012. An improved distance metric for the interpolation of link-based traffic data using kriging: a case study of a large-scale urban road network. Int. J. Geogr. Inf. Sci. 26, 667–689.

Appendices

Appendix A

Appendix A.1– Types of signal control

Type of control	Description
Isolated intersection control	Controls traffic without considering adjacent signalized intersections.
Interchange and closely spaced intersection control	Provides progressive traffic flow through two closely spaced intersections, such as interchanges. Control is typically done with a single traffic controller.
Arterial intersection control (open ne twork)	Provides progressive traffic flow along the arterial. This is accomplished by coordination of the traffic signals.
Closed network control	coordinates a group of adjacent signalized intersections.
Areawide system control	treats all or a major portion of signals in a city (or metropolitan area) as a total system. Isolated, open- or closed-network concepts may control individual signals within this area.

Source: Reproduced from Gordon and Tighe (2005**, p.53)**

Appendix A.2 – Signal timing variables

Variable	Definition
Cycle Length	The time required for one complete sequence of signal intervals (phases).
Phase	The portion of a signal cycle allocated to any single combination of one or more traffic movements simultaneously receiving the right-of-way during one or more intervals.
Interval	A discrete portion of the signal cycle during which the signal indications (pedestrian or vehicle) remain unchanged.
Split	The percentage of a cycle length allocated to each of the various phases in a signal cycle.
Offset	The time relationship, expressed in seconds or percent of cycle length, determined by the difference between a defined point in the coordinated green and a system reference point.

Source: Reproduced from Gordon and Tighe (2005**, p.67)**

Appendix B

Appendix B.1 – LOKRR training results for each of the CCZ links, start camera failure

Link	k	w	sigma	RMSE	NRMSE	MAPE	MASEAvg	%Patch
417	3	3	1	0.041059	0.092588	14.83272	0.956888	20.31073
418	1	1	3	0.063343	0.100798	22.70402	0.931543	19.03087
419	2	3	1	0.068813	0.059115	18.67571	0.992622	1.181421
420	4	3	1	0.082921	0.098975	23.8241	0.89607	17.20343
423	2	3	1	0.059325	0.061758	16.34856	0.783587	18.64379
425	2	3	3	0.03947	0.048386	15.50008	0.934568	2.039165
427	6	2	1	0.074605	0.077783	19.2696	1.010526	30.63063
432	8	3	1	0.033458	0.062696	14.84968	0.786516	7.460754
433	3	3	3	0.18031	0.158944	25.45947	0.898792	17.18725
435	6	3	1	0.071392	0.07776	16.18891	0.849838	6.554459
436	8	3	1	0.060489	0.092912	15.95847	0.968195	5.793818
437	2	3	3	0.146391	0.102853	21.54584	0.823277	24.89076
442	3	3	3	0.130491	0.127425	25.25464	0.93127	0.048552
446	8	3	1	0.315797	0.132521	34.93741	0.838869	3.65755
448	2	3	2	0.038544	0.132218	18.96833	1.014391	29.94012
453	3	3	3	0.056958	0.092268	16.62496	0.911202	0
454	7	3	1	0.146758	0.051134	22.21758	0.931852	0.825376
455	4	2	1	0.07381	0.081607	18.89547	0.885234	21.5251
457	1	1	1	0.067615	0.057801	26.21137	1.038959	0.2079
458	2	3	3	0.121145	0.064825	19.87544	0.853906	0.550251
459	1	3	2	0.059942	0.078355	21.40221	0.91832	7.120893
461	8	2	1	0.100028	0.106661	28.02516	0.883139	27.0592
463	3	3	1	0.231931	0.092234	26.50776	0.945296	19.27496
470	1	3	2	0.093426	0.02668	16.31714	0.955244	2.249555
472	1	3	1	0.095718	0.097338	25.48324	0.979563	0.598802
473	3	2	1	0.13407	0.08771	24.80363	0.898514	3.812741
474	1	3	1	0.113989	0.090896	36.59746	1.009552	25.86179
523	2	3	1	0.066795	0.056552	22.28719	0.847052	35.10277
524	3	3	1	0.119416	0.067627	23.64647	0.879511	33.69477
1396	7	3	1	0.085593	0.042849	19.44054	0.883277	15.52031
1400	5	3	1	0.033697	0.082675	13.52474	0.93244	10.37385
1412	1	3	3	0.05431	0.087393	19.96235	0.893463	33.93753
1413	3	3	1	0.185375	0.115139	33.93588	0.884782	19.22641
1419	2	3	1	0.112395	0.087174	19.95042	0.933318	2.071533
1421	3	3	1	0.035313	0.052132	15.55239	1.002187	1.812591
1453	4	3	1	0.098772	0.059112	19.74117	0.555926	2.233371
1518	5	3	3	0.023612	0.048076	12.20038	0.739128	1.035766
1520	6	3	1	0.042508	0.089136	12.64017	0.843162	9.273345
1576	4	3	1	0.089334	0.06651	19.5997	0.951155	15.1319

1593	1	3	1	0.208969	0.108945	24.18539	0.883821	23.19145
1604	1	2	1	0.08147	0.074784	22.35692	0.914813	4.343629
1613	4	3	2	0.067672	0.083923	17.78891	0.816468	5.421589
1614	1	3	3	0.075477	0.099758	41.56523	0.799644	3.851756
1616	2	3	2	0.112849	0.070002	17.56441	0.901011	27.94951
1632	3	3	3	0.0524	0.071073	17.32068	0.832143	16.41042
1745	8	2	3	0.094716	0.110708	39.61017	1.240681	4.005792
1747	5	3	1	0.075301	0.109416	20.69841	0.804152	6.942871
1810	4	3	1	0.112707	0.058856	21.33575	0.947036	24.27577
1882	7	3	1	0.144607	0.110594	29.7325	0.881138	9.467551
1883	3	3	1	0.049126	0.087101	16.75841	1.032691	2.6056
1887	4	3	1	0.049532	0.075148	23.18216	0.98259	12.28354
1951	6	3	1	0.111339	0.084183	20.92855	0.979144	0.048552
1956	3	3	1	0.083163	0.100776	23.79603	0.943881	0.679722
2043	6	3	1	0.086327	0.067317	20.24772	0.985964	2.913093
2044	1	1	1	0.086807	0.032853	36.55598	0.89828	19.3667
2054	5	3	1	0.058744	0.074437	16.5343	0.810639	0.906296
2059	1	3	3	0.045761	0.091305	16.10786	0.935934	10.34148
2063	3	3	1	0.067577	0.100022	26.36232	0.956042	17.91552
2086	2	3	3	0.026179	0.048873	10.31917	0.951232	5.76145
2088	4	3	3	0.027137	0.071939	12.47287	0.992576	22.60884
2089	7	3	1	0.038991	0.044607	16.39402	0.895691	19.97087
2090	5	3	2	0.0529	0.060683	13.09561	0.905995	9.62939
2102	3	3	1	0.082626	0.111817	22.19209	0.985306	42.06182
2109	8	3	1	0.096239	0.093866	21.26772	0.863689	2.94546
2112	2	1	1	0.035351	0.039251	12.82974	1.017589	1.135455
2318	3	3	1	0.063456	0.069018	18.53794	0.780465	7.622593
2361	2	3	1	0.061391	0.073997	17.37173	0.911125	13.48115
2363	1	3	1	0.027136	0.064109	14.73511	0.910628	0.291309
2364	1	3	3	0.131388	0.104952	26.82352	0.863757	25.53811
2432	5	3	1	0.077014	0.051379	24.92824	0.985069	4.304904
2433	1	2	2	0.101215	0.054666	24.21681	0.648635	26.43179
2445	3	3	1	0.089669	0.087194	18.62221	0.965444	1.586017
2447	5	3	1	0.149804	0.128704	26.1364	0.777369	0.097103
2450	1	3	1	0.061517	0.054607	19.17366	0.474851	13.09273
2451	4	3	1	0.131589	0.070209	24.54642	0.556092	17.05778
2453	4	3	3	0.035939	0.060739	12.75959	0.864283	5.114096
2454	3	1	3	0.052564	0.114909	19.79515	0.964513	26.19543
2461	5	3	2	0.053021	0.070807	14.76276	0.768504	5.405405
2462	2	3	1	0.10032	0.061067	22.65136	0.775745	30.3609
2466	1	3	1	0.19066	0.064026	26.81176	0.735634	10.53569

Appendix B.2 – LOKRR training results for each of the CCZ links, end camera failure

Link	k	w	sigma	RMSE	NRMSE	MAPE	MASEAvg	%Patch
417	3	3	1	0.0411	0.0926	14.8344	0.9571	20.3107
418	1	3	3	0.0543	0.0864	20.2357	0.8055	18.7247
419	3	3	1	0.0676	0.0581	17.8777	0.9609	1.1814
420	3	3	1	0.0835	0.0997	24.0212	0.9056	17.2034
423	2	3	1	0.0671	0.0699	17.4946	0.8701	18.6438
425	3	3	2	0.0396	0.0485	15.0592	0.9289	2.0392
427	3	3	1	0.0757	0.0789	18.0161	0.9806	30.4095
432	7	3	1	0.0397	0.0743	16.7639	0.8850	7.4608
433	4	3	1	0.1571	0.1385	22.1683	0.7629	17.1872
435	5	3	1	0.0711	0.0774	15.2165	0.8293	6.5545
436	6	3	1	0.0618	0.0950	15.4689	0.9642	5.7938
437	7	3	1	0.1201	0.0844	18.7025	0.6751	24.8908
442	7	3	1	0.1119	0.1093	22.5838	0.7970	0.0486
446	6	3	3	0.3847	0.1614	35.1484	1.0032	3.6575
448	1	3	3	0.0385	0.1321	18.9303	1.0131	29.9401
453	6	3	1	0.0574	0.0930	16.3865	0.8973	0.0000
454	1	1	3	0.1659	0.0578	21.3952	0.9753	0.8156
455	2	2	3	0.0687	0.0759	18.1525	0.8372	21.5251
457	2	1	1	0.0662	0.0566	25.6518	1.0228	0.2079
458	1	2	1	0.1395	0.0746	20.3373	0.9417	0.5952
459	3	3	1	0.0670	0.0876	22.8637	0.9894	7.1209
461	2	3	2	0.1085	0.1157	29.5584	0.9270	26.8814
463	1	3	3	0.2281	0.0907	25.2731	0.9097	19.2750
470	2	3	1	0.0946	0.0270	16.1141	0.9533	2.2496
472	2	3	1	0.1020	0.1037	26.0615	1.0224	0.5988
473	3	3	1	0.1367	0.0894	26.4893	0.9309	3.8356
474	1	3	3	0.1132	0.0902	35.5752	0.9947	25.8618
523	2	2	1	0.0668	0.0566	22.3626	0.8506	35.1190
524	4	3	1	0.1190	0.0674	23.5827	0.8772	33.6948
1396	8	3	1	0.0892	0.0446	20.6017	0.9221	15.5203
1400	3	3	1	0.0364	0.0894	14.4421	0.9897	10.3738
1412	1	3	3	0.0543	0.0874	19.9566	0.8932	33.9375
1413	3	3	1	0.1302	0.0808	20.9496	0.5810	19.2264
1419	1	3	3	0.0945	0.0733	17.8358	0.8198	2.0715
1421	2	3	2	0.0369	0.0545	16.0529	1.0279	1.8126
1453	2	3	3	0.1745	0.1044	33.1445	0.9903	2.2334
1518	1	3	1	0.0296	0.0603	15.3938	0.9096	1.0358
1520	4	3	2	0.0415	0.0870	12.8030	0.8466	9.2733
1576	4	3	1	0.0894	0.0665	19.5961	0.9510	15.1319
1593	1	3	1	0.2089	0.1089	24.1807	0.8836	23.1915
1604	1	2	1	0.0815	0.0748	22.3503	0.9146	4.3436
1613	5	3	1	0.0642	0.0797	16.4627	0.7624	5.4216

1614	3	2	1	0.0686	0.0907	34.5153	0.6897	3.9736
1616	1	3	2	0.1110	0.0688	17.5366	0.8844	27.9495
1632	3	3	1	0.0589	0.0799	18.9190	0.9175	16.4104
1745	2	2	2	0.0797	0.0932	24.5497	0.9038	4.0058
1747	8	3	1	0.0641	0.0932	19.9244	0.7211	6.9429
1810	3	3	1	0.1130	0.0590	21.6334	0.9568	24.2758
1882	3	3	1	0.1640	0.1255	30.8934	0.9396	9.4676
1883	3	3	1	0.0458	0.0812	15.8595	0.9803	2.6056
1887	1	2	3	0.0515	0.0781	24.0161	1.0165	12.3552
1951	4	3	1	0.1167	0.0883	19.0309	0.9719	0.0486
1956	3	3	1	0.0832	0.1008	23.7980	0.9438	0.6797
2043	5	3	1	0.0859	0.0670	20.1186	0.9792	2.9131
2044	1	3	1	0.0864	0.0327	36.5310	0.8926	18.9837
2054	2	3	1	0.0520	0.0659	16.1784	0.7507	0.9063
2059	1	3	3	0.0458	0.0913	16.0990	0.9356	10.3415
2063	1	3	1	0.0561	0.0830	20.8407	0.7834	17.9155
2086	2	3	2	0.0262	0.0489	10.3278	0.9494	5.7615
2088	3	3	3	0.0272	0.0721	12.7365	1.0003	22.6088
2089	1	1	1	0.0429	0.0490	18.8199	1.0226	20.0384
2090	7	3	1	0.0516	0.0591	13.3680	0.9114	9.6294
2102	3	3	1	0.0845	0.1144	21.6958	0.9843	42.0618
2109	6	3	1	0.0966	0.0942	20.8028	0.8577	2.9455
2112	2	1	1	0.0354	0.0393	12.8302	1.0178	1.1355
2318	3	3	1	0.0656	0.0714	20.5533	0.8420	7.6226
2361	2	3	1	0.0614	0.0740	17.3824	0.9117	13.4811
2363	1	3	1	0.0271	0.0641	14.7321	0.9104	0.2913
2364	3	2	1	0.1005	0.0801	19.3692	0.6286	25.4183
2432	5	3	1	0.0685	0.0457	19.8978	0.8205	4.3049
2433	2	3	2	0.1219	0.0659	34.2077	0.8365	26.3149
2445	3	3	3	0.0834	0.0811	17.8660	0.9315	1.5860
2447	2	3	2	0.1277	0.1097	24.3696	0.6816	0.0971
2450	2	3	1	0.0590	0.0523	18.2499	0.4585	13.0927
2451	3	3	2	0.1876	0.1001	35.8739	0.8094	17.0578
2453	3	2	1	0.0406	0.0685	13.6358	0.9519	5.1319
2454	6	3	2	0.0432	0.0945	17.6617	0.8125	26.3635
2461	3	3	1	0.0613	0.0819	18.0839	0.9145	5.4054
2462	3	3	1	0.1071	0.0652	22.6314	0.8078	30.3609
2466	2	3	1	0.1799	0.0604	25.6890	0.7055	10.5357

Appendix B.3 – LOKRR training results for each of the CCZ links, both camera failure

Link	k	w	sigma	RMSE	NRMSE	MAPE	MASEAvg	%Patch
417	3	3	1	0.041069	0.09261	14.83236	0.956896	20.31073
418	1	1	3	0.063357	0.10082	22.72062	0.931952	19.03087
419	2	3	1	0.068926	0.059212	18.6736	0.992759	1.181421
420	3	3	1	0.083537	0.09971	24.02396	0.905554	17.20343
423	3	3	1	0.066214	0.06893	17.44095	0.861372	18.64379
425	3	3	2	0.040517	0.049669	15.34456	0.943864	2.039165
427	6	2	1	0.074612	0.07779	19.26516	1.010387	30.63063
432	5	3	1	0.038596	0.072323	16.34844	0.859774	7.460754
433	3	3	3	0.180319	0.158952	25.45508	0.898749	17.18725
435	4	3	2	0.071708	0.078104	16.25343	0.856484	6.554459
436	8	3	1	0.059712	0.091719	15.88307	0.953148	5.793818
437	5	3	2	0.144001	0.101174	21.47436	0.808424	24.89076
442	5	2	2	0.130317	0.127255	24.74946	0.925676	0.048263
446	7	3	3	0.382069	0.160332	37.86569	1.031303	3.65755
448	1	3	3	0.038511	0.132104	18.93323	1.013123	29.94012
453	2	3	3	0.060094	0.097347	17.79533	0.947836	0
454	2	1	1	0.165894	0.057801	21.32451	0.974904	0.815609
455	4	2	1	0.073823	0.081621	18.89641	0.885381	21.5251
457	3	2	1	0.066613	0.056945	24.49401	0.996425	0.209138
458	1	2	1	0.139466	0.074579	20.34658	0.942007	0.595238
459	3	3	1	0.067057	0.087655	22.86941	0.989581	7.120893
461	2	3	2	0.108515	0.115711	29.56506	0.927152	26.88137
463	3	3	1	0.231876	0.092212	26.50649	0.945362	19.27496
470	2	3	1	0.094579	0.027009	16.11518	0.953322	2.249555
472	2	3	1	0.101991	0.103717	26.06583	1.022384	0.598802
473	2	3	1	0.138809	0.09081	26.77783	0.946685	3.835572
474	2	3	1	0.116717	0.093071	37.02589	1.027519	25.86179
523	2	2	1	0.066811	0.056566	22.36493	0.850708	35.11905
524	3	3	1	0.119465	0.067655	23.65043	0.879666	33.69477
1396	8	3	1	0.089058	0.044584	20.60693	0.921531	15.52031
1400	3	3	1	0.036418	0.089351	14.44381	0.989842	10.37385
1412	1	3	3	0.054264	0.08732	19.95268	0.893003	33.93753
1413	1	3	3	0.186607	0.115904	31.92742	0.875354	19.22641
1419	2	2	1	0.112375	0.087107	19.95659	0.935834	2.059202
1421	4	3	1	0.034915	0.051544	15.23057	0.988883	1.812591
1453	2	3	3	0.174519	0.104444	33.17313	0.990614	2.233371
1518	5	3	2	0.028026	0.057064	13.18833	0.808926	1.035766
1520	3	3	1	0.045742	0.095917	13.62316	0.910261	9.273345
1576	4	3	1	0.089376	0.066541	19.6009	0.951303	15.1319
1593	1	3	1	0.208947	0.108934	24.1773	0.883669	23.19145
1604	1	2	1	0.081465	0.074779	22.35664	0.914646	4.343629
1613	5	3	1	0.069908	0.086695	20.06642	0.876828	5.421589

1614	2	2	1	0.069241	0.091516	36.39259	0.713213	3.973616
1616	5	3	1	0.114683	0.07114	18.19345	0.941626	27.94951
1632	4	3	1	0.059548	0.080769	18.72938	0.912574	16.41042
1745	8	2	3	0.094639	0.110617	39.53508	1.239572	4.005792
1747	4	3	1	0.073169	0.106317	20.39086	0.788966	6.942871
1810	3	3	1	0.112931	0.058973	21.63456	0.956725	24.27577
1882	2	1	1	0.16594	0.12644	32.40375	0.956772	9.483448
1883	3	3	1	0.049125	0.087099	16.75797	1.032689	2.6056
1887	1	2	3	0.051466	0.078084	24.0165	1.016545	12.35521
1951	5	3	1	0.116722	0.088253	19.44335	0.981225	0.048552
1956	2	3	1	0.083181	0.100797	24.37298	0.953965	0.679722
2043	5	3	1	0.08586	0.066952	20.11748	0.979115	2.913093
2044	1	1	1	0.086822	0.032858	36.57013	0.898678	19.3667
2054	4	2	1	0.056886	0.072082	16.09346	0.792631	1.061776
2059	1	3	3	0.045752	0.091287	16.10011	0.935562	10.34148
2063	1	3	1	0.065415	0.096823	26.01184	0.938414	17.91552
2086	2	3	2	0.026215	0.048942	10.32814	0.950471	5.76145
2088	2	3	3	0.027453	0.072777	12.88104	1.014719	22.60884
2089	1	1	1	0.042882	0.049058	18.8746	1.025265	20.03838
2090	4	3	3	0.053202	0.06103	13.45905	0.923952	9.62939
2102	3	3	1	0.084544	0.114413	21.69272	0.984113	42.06182
2109	7	3	1	0.096637	0.094254	21.21265	0.866935	2.94546
2112	2	1	1	0.035347	0.039245	12.82384	1.01737	1.135455
2318	2	3	2	0.065902	0.071678	20.63858	0.844012	7.622593
2361	2	3	1	0.061417	0.074028	17.37544	0.911318	13.48115
2363	1	3	1	0.02713	0.064093	14.7302	0.910266	0.291309
2364	1	3	3	0.131435	0.10499	26.82079	0.86381	25.53811
2432	5	3	1	0.077	0.05137	24.93235	0.985234	4.304904
2433	2	3	1	0.119164	0.064361	31.85832	0.806637	26.31494
2445	1	2	3	0.091512	0.088987	18.88155	0.989379	1.608752
2447	4	3	1	0.151418	0.13009	27.0066	0.791541	0.097103
2450	2	1	2	0.060117	0.053365	18.72345	0.464357	13.35359
2451	4	3	3	0.198063	0.105677	31.93134	0.805902	17.05778
2453	4	2	1	0.040549	0.06853	13.65099	0.95249	5.131918
2454	5	2	1	0.052537	0.114848	20.61742	0.971143	26.31918
2461	2	3	1	0.060428	0.080698	17.35985	0.882312	5.405405
2462	3	3	1	0.107133	0.065214	22.6319	0.807979	30.3609
2466	1	3	1	0.19065	0.064022	26.79879	0.735126	10.53569

Appendix B.4 – LOKRR testing results for each of the CCZ links, start camera failure

Link	RMSE	NRMSE	MAPE	MASEAvg
1745	0.1145	0.0837	29.9774	1.0653
1747	0.0672	0.0947	23.2331	0.9547
1883	0.0686	0.0484	17.3283	1.0065
1951	0.0482	0.0542	17.4165	1.0696
2462	0.1072	0.0540	25.8846	0.9386
2466	0.1791	0.0586	30.2766	0.8482
417	0.0435	0.0713	12.5601	0.9798
418	0.0588	0.1313	23.5860	1.0618
419	0.0633	0.0469	17.2184	1.0710
420	0.1330	0.0742	31.3592	1.0039
423	0.0766	0.0437	17.3639	0.8999
1887	0.0717	0.0538	23.9167	1.0063
1576	0.1037	0.0704	20.7721	0.9278
1956	0.0778	0.0604	26.1237	1.1276
1593	0.2942	0.1223	28.4093	0.8577
1604	0.1726	0.1179	22.6782	1.1386
1810	0.0712	0.0759	20.9924	1.0331
1613	0.0866	0.0880	20.8847	0.9212
1614	0.0579	0.0671	29.2537	0.5863
1616	0.1827	0.0340	23.7617	1.0566
1882	0.0920	0.1003	23.8423	1.0336
1632	0.0628	0.0843	17.8177	0.9178
1453	0.0862	0.0351	17.1802	0.6343
1518	0.0334	0.0770	12.0960	0.8419
1520	0.0246	0.0733	10.6173	0.8155
1396	0.0865	0.0443	22.0641	0.9043
1400	0.0364	0.0341	12.6382	0.9792
1412	0.0803	0.0565	19.1768	0.9343
1413	0.2371	0.1201	35.4001	1.0465
1419	0.0884	0.0564	19.8912	1.0331
1421	0.0545	0.1171	20.0682	1.0126
2090	0.0421	0.0279	14.3242	1.0522
2054	0.0674	0.0945	16.6733	0.9593
2112	0.0423	0.0503	12.9291	0.9645
2364	0.1666	0.1016	24.4919	0.9251
2063	0.0554	0.0733	20.6865	0.8869
2059	0.0542	0.0597	18.2972	1.0161
2361	0.0606	0.0946	14.8068	1.0034
2363	0.0415	0.0493	13.8855	0.8656
2043	0.1273	0.0993	26.4883	1.0898
2044	0.1505	0.0595	28.4175	0.8955
2086	0.0453	0.0349	10.6891	0.9723

2318	0.0513	0.0738	18.4057	0.8965
2088	0.0315	0.0823	12.2085	1.0151
2089	0.0409	0.0951	15.3125	0.9093
2102	0.0648	0.0910	17.2124	0.9262
2109	0.1173	0.1045	27.4309	0.9076
2432	0.1102	0.0297	19.6636	1.0098
2433	0.1262	0.0372	23.5955	0.8412
523	0.0720	0.0361	18.7291	0.8699
433	0.1066	0.0841	18.7260	0.7245
435	0.0503	0.0433	15.8850	1.0277
436	0.0739	0.0583	16.6861	0.9457
524	0.1149	0.1018	23.5435	0.9090
2454	0.0771	0.0579	22.0857	0.8987
437	0.0789	0.0771	20.0440	0.8732
2461	0.1091	0.0872	19.4982	0.8639
470	0.0852	0.0526	15.3810	0.9614
472	0.2370	0.1263	26.9904	0.9596
473	0.1699	0.0453	25.0124	0.8994
474	0.1547	0.1058	28.0096	0.9394
442	0.1343	0.0463	28.9205	0.9389
446	0.1563	0.1147	32.8306	1.0827
448	0.0416	0.0848	17.9629	1.0128
425	0.0337	0.0712	15.9547	1.0669
427	0.0766	0.0495	19.5039	1.0287
2445	0.0889	0.0491	20.8876	0.9975
2447	0.1804	0.1673	33.9956	0.8741
453	0.0732	0.0356	17.6181	0.9958
454	0.1226	0.0770	23.0016	0.9258
455	0.0816	0.1077	23.3960	1.0503
457	0.1024	0.0570	19.2735	0.9486
2450	0.0642	0.0919	19.4923	0.5261
2451	0.1243	0.0719	25.4346	0.5683
432	0.0399	0.0269	13.5574	0.7372
458	0.0705	0.0380	20.7321	1.0684
459	0.0703	0.0681	21.2092	0.9580
2453	0.0415	0.0754	13.4057	0.8106
461	0.0764	0.1069	26.1190	0.9021
463	0.2645	0.0993	29.8104	0.9649

Appendix B.5 – LOKRR testing results for each of the CCZ links, end camera failure

Link	RMSE	NRMSE	MAPE	MASEAvg
417	0.043476	0.071264	12.56144	0.979862
418	0.053579	0.119581	20.0027	0.931406
419	0.059457	0.044028	16.12688	0.994008
420	0.128214	0.07153	31.56306	0.997138
423	0.082751	0.047244	17.62003	0.939688
425	0.032434	0.068539	14.61476	1.008849
427	0.080863	0.052215	20.06932	1.070253
432	0.041642	0.028052	14.17307	0.770994
433	0.089925	0.070943	14.07534	0.576381
435	0.051691	0.044503	16.57989	1.068176
436	0.076242	0.060136	16.1667	0.950813
437	0.067991	0.0665	17.32503	0.75843
442	0.110332	0.038062	24.56454	0.729511
446	0.225914	0.165856	62.54534	1.966651
448	0.041345	0.084179	18.02522	1.011519
453	0.071765	0.034889	16.67063	0.95477
454	0.143145	0.089952	21.43905	0.981518
455	0.070605	0.093202	19.13215	0.883249
457	0.10121	0.05639	18.68105	0.926672
458	0.067691	0.036469	20.28585	1.023741
459	0.072162	0.069873	21.75208	0.973955
461	0.075196	0.105206	25.41075	0.886866
463	0.252864	0.094964	25.76303	0.88782
470	0.105683	0.065198	17.69794	1.083487
472	0.240885	0.128407	27.56743	0.981115
473	0.182118	0.048573	24.02674	0.910708
474	0.155807	0.106512	26.85644	0.932799
523	0.072171	0.036197	18.72749	0.872062
524	0.116859	0.103531	23.69496	0.921789
1396	0.103654	0.053133	24.64245	1.065845
1400	0.038753	0.036347	13.00834	1.01286
1412	0.080328	0.056524	19.1756	0.93451
1413	0.165155	0.083637	21.60357	0.668788
1419	0.080719	0.051463	19.60052	0.965183
1421	0.053016	0.113826	20.85796	1.013475
1453	0.141443	0.057687	30.88903	1.055452
1518	0.036788	0.084766	12.95449	0.910168
1520	0.02287	0.068218	9.854175	0.765766
1576	0.103707	0.070407	20.76671	0.927566
1593	0.294189	0.122331	28.41155	0.857627
1604	0.172568	0.117889	22.67069	1.13825
1613	0.067799	0.068903	16.78857	0.722842

1614	0.059978	0.069487	29.60982	0.601801
1616	0.181094	0.033715	22.55553	1.034025
1632	0.060086	0.080712	16.80269	0.888654
1745	0.106851	0.078146	25.7737	0.961044
1747	0.06257	0.088117	21.96836	0.915689
1810	0.075375	0.080359	21.54777	1.066173
1882	0.087936	0.095864	24.66533	1.022427
1883	0.062383	0.043961	15.91462	0.923076
1887	0.074732	0.055997	23.65728	1.020749
1951	0.053194	0.059813	18.78072	1.160029
1956	0.077766	0.060426	26.11891	1.127458
2043	0.12276	0.095726	25.70968	1.054126
2044	0.150955	0.059638	28.44925	0.891511
2054	0.056729	0.07953	15.25441	0.832856
2059	0.054235	0.059667	18.29889	1.01629
2063	0.049204	0.065149	18.063	0.786878
2086	0.045936	0.03541	10.74348	0.977462
2088	0.032375	0.084422	12.4353	1.03393
2089	0.041496	0.096409	15.73248	0.936707
2090	0.044396	0.029419	14.53511	1.081004
2102	0.065855	0.09251	17.50653	0.939107
2109	0.110699	0.098608	24.79719	0.842667
2112	0.042228	0.050254	12.92056	0.963828
2318	0.064506	0.092796	22.2188	1.097049
2361	0.060592	0.094613	14.80016	1.003229
2363	0.041481	0.049336	13.88471	0.865588
2364	0.13441	0.081938	19.56767	0.731361
2432	0.109038	0.029379	18.09049	0.956089
2433	0.135831	0.039982	25.31172	0.909613
2445	0.085186	0.047034	20.53221	0.983334
2447	0.140806	0.130589	24.5873	0.673439
2450	0.058846	0.084305	18.29256	0.487691
2451	0.200588	0.11608	44.83581	0.945952
2453	0.050777	0.092242	15.31834	0.974299
2454	0.066118	0.049642	19.54276	0.755782
2461	0.092236	0.073712	20.17806	0.93028
2462	0.114903	0.057857	27.89779	0.997983
2466	0.174561	0.057085	29.58868	0.833969
	0.094046	0.072213	21.01788	0.940446

Appendix B.6 – LOKRR testing results for each of the CCZ links, both camera failure

Link	RMSE	NRMSE	MAPE	MASEAvg
417	0.0435	0.0713	12.5593	0.9797
418	0.0589	0.1313	23.5964	1.0623
419	0.0633	0.0469	17.2208	1.0710
420	0.1282	0.0715	31.5593	0.9969
423	0.0819	0.0468	17.4667	0.9309
425	0.0348	0.0735	15.6199	1.0719
427	0.0767	0.0495	19.4950	1.0285
432	0.0408	0.0275	13.8022	0.7514
433	0.1066	0.0841	18.7323	0.7246
435	0.0517	0.0445	16.1578	1.0473
436	0.0790	0.0623	16.9412	0.9868
437	0.0768	0.0751	19.6569	0.8562
442	0.1321	0.0456	29.5377	0.9325
446	0.2555	0.1876	68.2888	2.1251
448	0.0413	0.0842	18.0305	1.0117
453	0.0756	0.0367	17.3994	1.0059
454	0.1422	0.0893	24.5879	1.0364
455	0.0816	0.1078	23.4010	1.0509
457	0.1010	0.0563	18.6625	0.9223
458	0.0677	0.0365	20.2973	1.0240
459	0.0722	0.0699	21.7588	0.9741
461	0.0752	0.1052	25.4121	0.8868
463	0.2645	0.0993	29.8324	0.9651
470	0.1058	0.0652	17.6988	1.0834
472	0.2408	0.1284	27.5616	0.9809
473	0.1808	0.0482	23.9611	0.9096
474	0.1607	0.1098	28.2486	0.9716
523	0.0721	0.0362	18.7260	0.8720
524	0.1149	0.1018	23.5369	0.9090
1396	0.1039	0.0532	24.6630	1.0672
1400	0.0389	0.0365	13.0140	1.0138
1412	0.0803	0.0565	19.1688	0.9341
1413	0.2262	0.1146	29.0842	0.9271
1419	0.0887	0.0565	19.8501	1.0339
1421	0.0528	0.1134	20.0749	0.9946
1453	0.1414	0.0577	30.8617	1.0545
1518	0.0355	0.0818	12.4696	0.8732
1520	0.0245	0.0730	10.2802	0.7993
1576	0.1037	0.0704	20.7681	0.9275
1593	0.2942	0.1223	28.4159	0.8578
1604	0.1726	0.1179	22.6801	1.1386
1613	0.0829	0.0843	20.3893	0.9027

1614	0.0583	0.0676	29.7703	0.5971
1616	0.1851	0.0345	24.6701	1.0977
1632	0.0647	0.0869	17.6259	0.9370
1745	0.1145	0.0838	29.9761	1.0654
1747	0.0650	0.0916	23.1197	0.9453
1810	0.0754	0.0804	21.5481	1.0663
1882	0.0866	0.0944	25.0493	1.0273
1883	0.0686	0.0484	17.3291	1.0066
1887	0.0747	0.0560	23.6563	1.0206
1951	0.0521	0.0586	18.4901	1.1460
1956	0.0776	0.0603	26.0783	1.1279
2043	0.1228	0.0957	25.7110	1.0545
2044	0.1505	0.0595	28.4190	0.8955
2054	0.0672	0.0942	16.5152	0.9523
2059	0.0542	0.0597	18.2885	1.0157
2063	0.0550	0.0728	20.9390	0.8907
2086	0.0458	0.0353	10.7146	0.9763
2088	0.0314	0.0818	12.0926	1.0089
2089	0.0420	0.0975	15.8882	0.9486
2090	0.0426	0.0282	14.3524	1.0583
2102	0.0659	0.0926	17.5095	0.9394
2109	0.1158	0.1032	26.9282	0.8926
2112	0.0422	0.0502	12.9208	0.9639
2318	0.0626	0.0900	21.6144	1.0710
2361	0.0605	0.0944	14.7798	1.0019
2363	0.0415	0.0493	13.8847	0.8656
2364	0.1666	0.1016	24.4975	0.9252
2432	0.1102	0.0297	19.6664	1.0099
2433	0.1318	0.0388	25.1212	0.8888
2445	0.0942	0.0520	21.3670	1.0527
2447	0.1819	0.1687	34.8787	0.8848
2450	0.0629	0.0664	18.8531	0.5063
2451	0.1941	0.1123	40.2472	0.8773
2453	0.0505	0.0917	15.2751	0.9689
2454	0.0753	0.0565	22.4061	0.8860
2461	0.0945	0.0755	20.0801	0.9466
2462	0.1149	0.0578	27.9015	0.9980
2466	0.1791	0.0586	30.2870	0.8483

Appendix B.7 – LOKRR training results for all CCZ links, with local STN

	Scenario 1				Scenario 2				Scenario 3			
Link	RMSE	NRMSE	MAPE	MASE Avg	RMSE	NRMSE	MAPE	MASE Avg	RMSE	NRMSE	MAPE	MASE Avg
1745	0.087121	0.024605	35.24432	1.010218	0.077097	0.024605	25.53307	0.815004	0.086914	0.024605	35.16104	1.008046
1747	0.072502	0.020476	20.01448	0.736763	0.064958	0.020476	19.75756	0.690702	0.071257	0.020476	21.04276	0.749963
1883	0.04759	0.01344	16.03224	0.944624	0.042877	0.01344	14.97264	0.879663	0.04777	0.01344	16.11096	0.94874
1951	0.107871	0.030465	19.34232	0.727368	0.115358	0.030465	18.83334	0.754255	0.114561	0.030465	19.51774	0.76359
2462	0.097143	0.027435	21.96195	0.634858	0.105193	0.027435	22.20691	0.667878	0.10534	0.027435	22.20478	0.667099
2466	0.183718	0.051885	26.35088	0.666254	0.172307	0.051885	25.79623	0.645579	0.183573	0.051885	26.4082	0.666891
417	0.039694	0.01121	14.41815	0.729383	0.039708	0.01121	14.39879	0.728919	0.039549	0.01121	14.38452	0.727802
418	0.062083	0.017533	22.05632	0.827918	0.052022	0.017533	19.38974	0.709212	0.062139	0.017533	22.07341	0.828928
419	0.06664	0.01882	17.8147	0.780274	0.065734	0.01882	17.32846	0.762794	0.066452	0.01882	17.80771	0.778515
420	0.080087	0.022618	23.01559	0.755205	0.080174	0.022618	22.99363	0.756831	0.080317	0.022618	23.02429	0.758469
423	0.056476	0.01595	15.88178	0.70759	0.063368	0.01595	16.86554	0.78084	0.063522	0.01595	17.10332	0.785788
1887	0.047692	0.013469	22.22712	0.917772	0.049636	0.013469	23.01567	0.944743	0.049579	0.013469	23.07333	0.945776
1576	0.087547	0.024725	18.76113	0.807482	0.087503	0.024725	18.7382	0.806112	0.087625	0.024725	18.77763	0.8081
1956	0.077648	0.021929	21.96084	0.751742	0.078912	0.021929	22.25111	0.758646	0.07904	0.021929	22.25992	0.759425
1593	0.202424	0.057168	23.84187	0.679222	0.205595	0.057168	23.83138	0.688238	0.205651	0.057168	24.29005	0.694221
1604	0.079164	0.022358	21.72037	0.695147	0.079168	0.022358	21.75598	0.6959	0.079012	0.022358	21.74821	0.694645
1810	0.109182	0.030835	20.83124	0.866043	0.110441	0.030835	20.99096	0.873564	0.110344	0.030835	20.9544	0.871795
1613	0.064525	0.018223	18.03705	0.763628	0.061009	0.018223	16.24124	0.707223	0.066766	0.018223	18.76616	0.791767
1614	0.069824	0.01972	37.79942	1.14955	0.064962	0.01972	31.82147	1.005537	0.065176	0.01972	32.46029	1.019164
1616	0.110376	0.031172	16.97198	0.776026	0.107121	0.031172	16.78975	0.755668	0.11064	0.031172	17.258	0.790274
1882	0.12741	0.035983	27.21359	0.677952	0.155093	0.035983	29.39214	0.766354	0.155321	0.035983	29.00942	0.763623
1632	0.051027	0.014411	16.96827	0.706384	0.056918	0.014411	18.25038	0.771179	0.057734	0.014411	18.46417	0.778972
1453	0.093632	0.026443	19.19939	0.434427	0.170044	0.026443	34.07215	0.806637	0.170011	0.026443	34.0421	0.80634
1518	0.022325	0.006305	11.5678	0.774672	0.026898	0.006305	12.65697	0.858714	0.026656	0.006305	12.81038	0.861978
1520	0.041705	0.011778	12.34921	0.858285	0.040085	0.011778	12.46593	0.858765	0.045159	0.011778	13.27408	0.928933
1396	0.083735	0.023648	18.81901	0.794317	0.087386	0.023648	19.70562	0.828223	0.087158	0.023648	19.72637	0.82822
1400	0.032697	0.009234	12.97109	0.805726	0.034507	0.009234	13.64086	0.843488	0.034429	0.009234	13.63936	0.843448
1412	0.052123	0.014721	19.7544	0.686306	0.052069	0.014721	19.48977	0.681453	0.052211	0.014721	19.46427	0.680856
1413	0.177529	0.050138	32.2791	0.696987	0.12849	0.050138	21.28007	0.480762	0.178431	0.050138	31.93384	0.705315
1419	0.110308	0.031153	19.56834	0.824409	0.093775	0.031153	17.86934	0.737861	0.110367	0.031153	19.57155	0.824864
1421	0.03346	0.00945	14.78521	0.760901	0.03413	0.00945	14.87916	0.769657	0.033586	0.00945	14.78688	0.762833

2090	0.052288	0.014767	13.10706	0.87154	0.051018	0.014767	13.04413	0.861616	0.052482	0.014767	13.38184	0.8844
2054	0.055499	0.015674	16.53819	0.720151	0.050539	0.015674	15.49742	0.662404	0.054503	0.015674	15.87018	0.701257
2112	0.034125	0.009638	12.02475	0.71823	0.033879	0.009638	11.82137	0.71021	0.034137	0.009638	12.06667	0.720434
2364	0.125938	0.035567	26.56833	0.678861	0.098028	0.035567	19.27096	0.50595	0.126257	0.035567	26.53977	0.678383
2063	0.062694	0.017706	25.06866	0.903738	0.051707	0.017706	19.54771	0.733847	0.061983	0.017706	24.88324	0.895704
2059	0.044626	0.012603	15.50571	0.774775	0.044238	0.012603	15.61071	0.77709	0.044538	0.012603	15.5607	0.777267
2361	0.057901	0.016352	15.93347	0.669547	0.057983	0.016352	15.94496	0.669967	0.057934	0.016352	15.89197	0.668805
2363	0.02538	0.007168	13.36694	0.876877	0.026261	0.007168	13.82222	0.898649	0.026289	0.007168	13.8504	0.900147
2043	0.084933	0.023987	19.9212	0.935903	0.084376	0.023987	19.87136	0.930059	0.084355	0.023987	19.86965	0.929867
2044	0.08568	0.024198	35.76827	0.983868	0.085647	0.024198	35.69158	0.981208	0.085665	0.024198	35.75726	0.984293
2086	0.025576	0.007223	10.01437	0.896345	0.025483	0.007223	10.06135	0.89519	0.025598	0.007223	10.05422	0.896856
2318	0.059351	0.016762	18.13437	0.69804	0.061211	0.016762	19.84022	0.744633	0.06216	0.016762	20.0138	0.75311
2088	0.026277	0.007421	12.038	0.896788	0.026399	0.007421	12.26349	0.901244	0.026658	0.007421	12.23332	0.905478
2089	0.037237	0.010516	15.86086	1.010255	0.041424	0.010516	17.53749	1.120207	0.041597	0.010516	17.52062	1.118873
2102	0.080319	0.022684	21.61692	0.905212	0.083176	0.022684	21.7072	0.921343	0.083436	0.022684	21.56247	0.920121
2109	0.092027	0.02599	20.38443	0.858809	0.092256	0.02599	20.0858	0.851305	0.092229	0.02599	20.32409	0.858693
2432	0.074974	0.021174	24.23398	0.784465	0.065321	0.021174	19.50895	0.648375	0.07511	0.021174	24.16894	0.782803
2433	0.09725	0.027465	23.46711	0.631566	0.117339	0.027465	31.55654	0.800457	0.115814	0.027465	30.44296	0.782033
523	0.064518	0.018221	21.23322	0.825467	0.064429	0.018221	21.27233	0.826247	0.064633	0.018221	21.28448	0.82723
433	0.179002	0.050554	25.01342	1.042757	0.157277	0.050554	22.34341	0.904494	0.176929	0.050554	24.6373	1.024483
435	0.069918	0.019746	15.73857	0.831805	0.068955	0.019746	15.29359	0.814477	0.07018	0.019746	16.02371	0.842074
436	0.058925	0.016641	15.29916	0.830588	0.060139	0.016641	15.29451	0.839918	0.058108	0.016641	15.2214	0.819164
524	0.116858	0.033003	23.33022	0.795003	0.116046	0.033003	23.25632	0.789154	0.116939	0.033003	23.31844	0.795016
2454	0.050308	0.014208	19.8534	0.888023	0.041233	0.014208	17.28204	0.749882	0.050243	0.014208	19.78778	0.885741
437	0.142647	0.040286	21.2948	0.835953	0.115585	0.040286	18.17711	0.677578	0.141106	0.040286	21.18372	0.826595
2461	0.050665	0.014309	14.40201	0.762339	0.055658	0.014309	16.37711	0.851062	0.05769	0.014309	17.05947	0.881681
470	0.090334	0.025512	15.09754	0.799563	0.091842	0.025512	15.30041	0.813021	0.091868	0.025512	15.31406	0.813651
472	0.088558	0.02501	22.41034	0.625939	0.097006	0.02501	24.68473	0.685606	0.096916	0.02501	24.70029	0.685465
473	0.124623	0.035196	22.8909	0.678169	0.132309	0.035196	25.97438	0.743922	0.133558	0.035196	25.73739	0.747671
474	0.1067	0.030134	32.79214	0.792641	0.108978	0.030134	34.40257	0.823394	0.113499	0.030134	35.78016	0.851596
442	0.128306	0.036236	25.0994	0.846022	0.105561	0.036236	21.11428	0.685997	0.128197	0.036236	24.87529	0.841208
446	0.289425	0.081739	31.54973	0.748645	0.379638	0.081739	33.82375	0.960125	0.375167	0.081739	35.08706	0.964978
448	0.035915	0.010143	17.9561	0.878813	0.036253	0.010143	18.01142	0.885385	0.03632	0.010143	18.03879	0.887373
425	0.037318	0.010539	14.59478	0.780502	0.037416	0.010539	14.42296	0.777009	0.038281	0.010539	14.70806	0.794113
42	0.0710	0.0200	17.623	0.8808	0.0739	0.0200	17.532	0.8969	0.0710	0.0200	17.639	0.8810

7	33	61	14	97	24	61	47	78	24	61	99	03
2445	0.088172	0.024901	18.88246	0.844063	0.080912	0.024901	17.71343	0.802362	0.08955	0.024901	18.55173	0.847994
2447	0.14616	0.041278	25.9653	0.717087	0.122459	0.041278	23.36639	0.610391	0.147806	0.041278	26.56003	0.723668
453	0.054478	0.015386	16.62208	0.79936	0.055798	0.015386	16.52635	0.798842	0.05884	0.015386	17.15094	0.83091
454	0.142188	0.040157	21.37662	0.818771	0.163606	0.040157	21.22458	0.877362	0.165127	0.040157	21.60922	0.888834
455	0.071634	0.020231	18.75789	0.749359	0.066554	0.020231	17.69809	0.70631	0.071733	0.020231	18.77691	0.750065
457	0.065821	0.018589	24.42238	0.923243	0.064311	0.018589	23.66775	0.897765	0.062913	0.018589	23.81714	0.90396
2450	0.05849	0.016519	18.49714	0.677241	0.055286	0.016519	17.41558	0.639304	0.056935	0.016519	17.73135	0.652275
2451	0.120847	0.03413	23.80189	0.486471	0.182666	0.03413	35.33961	0.731633	0.194383	0.03413	31.91141	0.733821
432	0.031514	0.0089	14.20003	0.920669	0.037186	0.0089	16.11728	1.04438	0.035839	0.0089	15.33273	1.00018
458	0.120874	0.034137	19.68475	0.776968	0.138857	0.034137	20.54895	0.865752	0.13886	0.034137	20.54464	0.865313
459	0.057553	0.016254	21.49665	0.753951	0.065626	0.016254	22.66767	0.815136	0.065521	0.016254	22.61556	0.813844
2453	0.034311	0.00969	12.39173	0.693387	0.039507	0.00969	13.59213	0.777437	0.039526	0.00969	13.50313	0.775286
461	0.091611	0.025873	25.07963	0.845151	0.105146	0.025873	29.09283	0.972754	0.105041	0.025873	28.98278	0.970556
463	0.225166	0.063591	26.22372	0.78993	0.218125	0.063591	25.63601	0.766132	0.224215	0.063591	26.04014	0.783887
Avg	0.08389	0.023692	20.28616	0.792515	0.085464	0.023692	20.18834	0.793425	0.089425	0.023692	21.05833	0.823907

Appendix B.8 – LOKRR testing results for all CCZ links, with local STN

	Scenario 1				Scenario 1				Scenario 1			
Link	RMSE	NRMSE	MAPE	MASE Avg	RMSE	NRMSE	MAPE	MASE Avg	RMSE	NRMSE	MAPE	MASE Avg
1745	0.120073	0.094389	31.97451	1.181587	0.118583	0.09352	30.19908	1.138454	0.120887	0.088412	32.27236	1.187998
1747	0.072555	0.112424	23.29274	1.036544	0.073602	0.103697	22.57261	1.027812	0.074705	0.115809	23.96887	1.073743
1883	0.083566	0.05889	17.86987	1.099675	0.081838	0.057671	17.06931	1.066332	0.082687	0.05827	17.85597	1.111859
1951	0.056527	0.063868	18.12232	0.860463	0.06225	0.081488	21.24616	1.009949	0.062898	0.071296	20.86783	0.997568
2462	0.123322	0.062096	26.99865	0.976629	0.134629	0.067789	30.36476	1.079799	0.135035	0.067994	30.37201	1.080519
2466	0.226615	0.074107	34.26023	1.012154	0.231417	0.075678	34.25809	1.02119	0.226895	0.074199	34.04181	1.006538
417	0.04975	0.082132	12.82803	1.131163	0.049563	0.081824	12.8252	1.130481	0.04937	0.081504	12.78354	1.128699
418	0.058557	0.135137	22.29945	1.1202	0.053232	0.119411	19.58441	0.993078	0.05868	0.135423	22.25949	1.120353
419	0.069253	0.051282	17.75196	1.120697	0.067746	0.050258	16.99262	1.08528	0.069286	0.051307	17.8345	1.117588
420	0.142415	0.079453	28.85775	1.047725	0.136625	0.077077	29.15594	1.049414	0.135861	0.076646	29.11744	1.04891
423	0.096856	0.055297	17.85363	1.016342	0.100116	0.057121	18.24136	1.026611	0.10051	0.057346	18.30796	1.033118
1887	0.08687	0.065093	25.32333	1.163224	0.08257	0.063118	25.20965	1.154455	0.082767	0.063268	25.22872	1.150422
1576	0.116428	0.090302	21.22642	1.086036	0.116549	0.090396	21.24857	1.085531	0.117456	0.091099	21.32448	1.089663
1956	0.082822	0.064355	23.80038	0.99311	0.083986	0.065259	24.5755	1.016378	0.084542	0.065691	24.6943	1.02085
1593	0.298222	0.123225	28.25828	1.032356	0.290217	0.119917	27.77018	1.006144	0.293525	0.121284	28.24676	1.024115
1604	0.161944	0.111068	20.9303	1.054524	0.161815	0.110979	20.94637	1.047821	0.162211	0.11125	21.05463	1.05179
1810	0.082949	0.090856	21.91343	1.235491	0.079807	0.087414	21.41971	1.202548	0.080444	0.088112	21.53103	1.211551
1613	0.098588	0.100192	21.79272	1.109633	0.088617	0.090424	17.94098	0.92085	0.097537	0.104324	21.84831	1.112707
1614	0.051091	0.05919	22.92249	0.850031	0.05585	0.064704	24.35367	0.885707	0.058698	0.077363	24.8369	0.915454
1616	0.245871	0.045775	25.42966	1.220801	0.245432	0.045693	24.72488	1.211275	0.240935	0.044856	25.78767	1.2502
1882	0.087843	0.096016	22.48613	1.070214	0.093629	0.102431	24.14132	1.142399	0.095912	0.104559	24.44981	1.151078
1632	0.072658	0.099195	19.52975	1.114654	0.073112	0.098209	18.1437	1.059492	0.072063	0.096801	19.16347	1.108627
1453	0.106131	0.044806	16.88492	0.642803	0.17423	0.071259	35.30862	1.231922	0.174528	0.07138	35.27126	1.234257
1518	0.04275	0.098504	13.88779	1.174282	0.041263	0.095078	14.0825	1.157821	0.043423	0.11481	14.87239	1.235785
1520	0.026282	0.091806	11.04983	1.088712	0.026839	0.093749	10.89051	1.0711	0.028578	0.086212	11.42814	1.135269
1396	0.105778	0.054222	21.7836	0.996681	0.11647	0.059703	24.29446	1.139826	0.116284	0.059607	24.13062	1.134971
1400	0.05826	0.063945	14.54388	1.215076	0.061246	0.067222	14.9448	1.257169	0.060322	0.066208	14.92915	1.251263
1412	0.084364	0.059365	20.17334	1.092941	0.085356	0.060063	20.11585	1.095735	0.085058	0.059853	20.35592	1.106206
1413	0.260506	0.134415	35.20083	1.09758	0.213844	0.108293	25.63902	0.829714	0.269511	0.137959	35.24686	1.11726
1419	0.102112	0.065072	20.71324	1.082905	0.098697	0.066239	20.08253	1.014887	0.101934	0.064958	20.69721	1.08033
1421	0.057352	0.124269	20.26512	1.106087	0.058192	0.124937	21.42911	1.136632	0.056123	0.123312	20.57753	1.102768

2090	0.057059	0.03781	16.01763	1.297403	0.059827	0.039645	16.21378	1.328075	0.056924	0.037721	15.86898	1.274879
2054	0.069216	0.097036	17.39736	1.115115	0.062237	0.087253	15.84229	1.007139	0.068483	0.096009	17.20527	1.105216
2112	0.056158	0.070505	14.40723	1.139072	0.056979	0.068756	14.78524	1.165658	0.056241	0.07061	14.46105	1.138898
2364	0.184185	0.112282	27.0463	1.093572	0.17439	0.109614	21.7706	0.933195	0.185283	0.112951	27.1659	1.099065
2063	0.057126	0.075638	19.02639	1.021869	0.056518	0.074833	17.66092	0.957147	0.058791	0.077842	19.10464	1.024564
2059	0.070423	0.081125	20.55829	1.171418	0.069652	0.076629	19.17995	1.103171	0.070894	0.081667	20.89767	1.186013
2361	0.061232	0.095613	13.89035	1.12453	0.061346	0.095791	13.87015	1.124369	0.0617	0.096344	13.95767	1.129904
2363	0.046417	0.056054	14.97813	1.042786	0.045031	0.053557	13.92287	0.976627	0.045029	0.053555	13.98159	0.976945
2043	0.117321	0.091485	26.23388	1.122505	0.110181	0.085917	25.45724	1.086681	0.110611	0.086253	25.45353	1.084846
2044	0.181627	0.071756	33.16451	1.066136	0.174523	0.069234	33.36331	1.074852	0.182216	0.071988	32.88843	1.065087
2086	0.067247	0.054659	12.09342	1.106392	0.067826	0.055129	12.03063	1.111943	0.06862	0.055774	12.13336	1.134317
2318	0.060884	0.087585	18.56907	1.036797	0.067292	0.097557	21.4236	1.184819	0.065722	0.094545	20.85685	1.163424
2088	0.038045	0.099752	12.97676	1.176333	0.038935	0.116079	13.18649	1.202963	0.035909	0.094152	12.58961	1.135741
2089	0.04692	0.10901	14.68119	1.096574	0.047983	0.111481	14.94466	1.11377	0.045354	0.105526	14.43767	1.075364
2102	0.067718	0.095126	17.22823	1.01459	0.068252	0.095876	17.36563	1.011769	0.068212	0.09582	17.43497	1.012579
2109	0.122538	0.109154	28.21107	1.106233	0.119566	0.106932	27.41725	1.074477	0.120069	0.106955	27.52407	1.082759
2432	0.173195	0.046666	21.33901	1.094881	0.172785	0.046555	20.21763	1.051884	0.172522	0.046484	21.32164	1.092591
2433	0.136382	0.040001	23.55688	1.010486	0.136653	0.065338	26.55981	1.155559	0.127647	0.061032	26.17165	1.109529
523	0.079227	0.039761	19.14093	0.990303	0.078113	0.039202	19.20139	0.992221	0.076547	0.038416	19.09425	0.986149
433	0.102749	0.081061	19.80366	1.050928	0.08938	0.070514	14.73023	0.834342	0.10596	0.08363	20.38533	1.08806
435	0.062415	0.053736	16.76795	1.186742	0.061177	0.05267	16.41619	1.163215	0.060149	0.051785	16.9578	1.191345
436	0.088893	0.072099	18.04062	1.075521	0.090954	0.07174	17.98457	1.095577	0.094046	0.074178	18.74645	1.11769
524	0.122054	0.110369	25.06652	1.115676	0.124071	0.11056	25.09858	1.126646	0.122114	0.110424	25.10222	1.124649
2454	0.087648	0.065806	24.77449	1.051938	0.087889	0.065987	23.22792	0.98657	0.090186	0.067712	24.93902	1.054992
437	0.08818	0.101785	20.73573	1.04424	0.075542	0.075249	17.5364	0.875635	0.087153	0.092949	20.82043	1.042188
2461	0.10636	0.085261	19.63887	0.98637	0.108864	0.086773	21.90146	1.089549	0.10601	0.08472	21.80313	1.074911
470	0.102036	0.063199	17.16781	1.078647	0.109332	0.073272	18.64593	1.178882	0.109737	0.073543	18.68389	1.169843
472	0.240458	0.128179	29.72923	1.068888	0.240575	0.128242	28.27448	1.049167	0.242159	0.129087	28.2802	1.049726
473	0.207374	0.055309	27.43485	1.063499	0.219958	0.058666	26.2808	1.072582	0.22042	0.058789	26.67589	1.088167
474	0.167838	0.114737	28.79393	1.08806	0.15884	0.112128	26.64512	1.030122	0.162679	0.11121	27.51225	1.046409
442	0.146529	0.05055	34.57989	1.080736	0.125093	0.043254	25.9698	0.82638	0.148132	0.051221	34.68983	1.082993
446	0.132775	0.097477	29.17108	0.979194	0.216437	0.165486	56.72924	1.754096	0.229066	0.168054	59.70428	1.815105
448	0.041819	0.085142	16.60933	1.081995	0.041542	0.08458	16.84387	1.081453	0.041669	0.084837	16.92808	1.084703
425	0.036242	0.076584	15.60358	1.161075	0.036609	0.083663	15.64694	1.175895	0.037425	0.079084	15.71111	1.197252
427	0.0911	0.0693	21.383	1.0911	0.0925	0.0704	22.261	1.1316	0.0911	0.0693	21.348	1.0864

	47	82	77	09	19	26	56	25	41	77	87	12
2445	0.100131	0.055286	21.02379	1.051489	0.097823	0.054012	22.59162	1.097858	0.10404	0.057603	23.54484	1.167083
2447	0.180648	0.167694	37.2004	1.067753	0.149642	0.138783	29.58088	0.86826	0.186134	0.172787	38.22268	1.091971
453	0.084029	0.040851	18.64612	1.051903	0.083059	0.040379	18.99706	1.063674	0.083727	0.040704	19.46225	1.098214
454	0.124457	0.078208	24.19848	0.995631	0.15123	0.096484	30.35868	1.228246	0.152749	0.096413	29.23814	1.2037
455	0.087897	0.116162	21.33544	1.132034	0.084077	0.110987	20.04678	1.060924	0.088227	0.116598	21.41163	1.129467
457	0.107765	0.060041	19.75231	1.033222	0.107814	0.060069	19.30377	1.022059	0.10479	0.058384	19.41871	1.026253
2450	0.06348	0.052694	18.83359	0.815816	0.061549	0.067667	19.14942	0.819397	0.061859	0.090365	19.3434	0.834367
2451	0.145299	0.084084	27.66719	0.613539	0.2135	0.124509	43.77409	0.914775	0.214969	0.124402	43.04611	0.918243
432	0.051018	0.094857	15.75712	1.165875	0.054332	0.037088	16.05794	1.190796	0.050227	0.096387	15.67945	1.164259
458	0.064612	0.038801	20.5058	0.992577	0.074969	0.045021	21.63306	1.056612	0.072963	0.043816	21.6737	1.062603
459	0.070362	0.068129	21.21079	0.992504	0.078935	0.07643	22.41624	1.05371	0.078968	0.076462	22.51824	1.049328
2453	0.055288	0.100436	16.28621	1.052762	0.057338	0.115354	17.43961	1.144317	0.056995	0.114665	17.43061	1.140458
461	0.069295	0.09695	24.33595	0.966032	0.073674	0.104849	26.11722	1.028437	0.073838	0.105082	26.11472	1.030419
463	0.29925	0.117304	33.76512	1.11256	0.293644	0.115587	33.03442	1.094125	0.295209	0.111606	33.61366	1.102743